

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено
Завідувач кафедри

О.В. Коваль
(підпис) (ініціали, прізвище)

“ ” 2019р.

ДИПЛОМНА РОБОТА
на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 “Комп’ютерні науки”

на тему «Моніторинг нафтових розливів на морській поверхні методом
дистанційного зондування Землі»

Виконав: студент 4-го курсу, групи ТМ-52

Панченко Богдан Тарасович
(прізвище, ім’я, по батькові)
(підпис)

Керівник асист. Швайко В.Г.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
(підпис)

Рецензент доцент, к.т.н Баранюк О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
(підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.В. Коваль

(підпис)

” ____ ” _____ 2019р.

ЗАВДАННЯ

на дипломну роботу студенту

Панченку Богдану Тарасовичу

(прізвище, ім’я, по батькові)

1. Тема роботи Моніторинг нафтових розливів на морській поверхні методом дистанційного зондування Землі

керівник роботи _____ асист. Швайко Валерій Григорович

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”22” 03 2018р. №1004-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи мова програмування C#, платформа Microsoft Visual Studio 2015

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програмні засоби, розробити застосунок, спроектувати структуру бази даних, спроектувати архітектуру програми, реалізувати інтерфейс користувача системи, розробити інструкцію користувача

5. Перелік ілюстративного матеріалу

Мета розробки, Актуальність, Фільтр Гауса, Алгоритм фільтрації зображення, Засоби розробки, Зображення оброблених знімків, Інтерфейс користувача

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| | | | |

7. Дата видачі завдання ” 1 ” грудня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання дипломної роботи | Термін виконання етапів роботи | Примітки |
|-------|---|--------------------------------|----------|
| 1. | Затвердження теми роботи | 01.12.2018 | |
| 2. | Вивчення та аналіз задачі | 02.12.2018 – 19.12.2018 | |
| 3. | Розробка архітектури та загальної структури системи | 19.12.2018 – 29.12.2018 | |
| 4. | Розробка структур окремих підсистем | 09.01.2019 – 23.01.2019 | |
| 5. | Програмна реалізація системи | 09.01.2019 – 19.01.2019 | |
| 6. | Оформлення пояснювальної записки | 22.01.2019 – 12.03.2019 | |
| 7. | Захист програмного продукту | 17.05.2019 | |
| 8. | Передзахист | 31.05.2019 | |
| 9. | Захист | 17.06.2019 – 21.06.2019 | |

Студент _____ Панченко Б.Т.
(підпис) (прізвище та ініціали,)

Керівник роботи _____ Швайко В.Г.
(підпис) (прізвище та ініціали,)

ABSTRACT

Diploma work is devoted to the developing of a tool for monitoring the oil spills on the sea surface by methods of remote sensing of the Earth using the platform .NET Framework with the use of ArcObject SDK libraries. Developed system is created for use in relevant institutions involved in monitoring and eliminating oil pollution of the seas.

The paper consists of 5 sections, 70 pages, 20 pictures, 4 tables, 3 additions and 15 sources.

АНОТАЦІЯ

Дипломна робота присвячена розробці інструментального засобу моніторингу нафтових розливів на морській поверхні методами дистанційного зондування Землі засобами платформи .NET Framework з застосуванням бібліотек ArcObject SDK. Розроблена система призначається для використання у відповідних закладах , які займаються моніторингом та усуненням нафтових забруднень морів.

Записка складається з 5 розділів, 70 сторінок, 20 рисунків, 4 таблиць, 3 додатків та 15 використаних джерел.

Розроблене програмне забезпечення дозволяє визначати межі нафтового забруднення морської поверхні. Програма проводить обробку результатів дистанційного зондування Землі і повертає результат у вигляді растрового зображення.

ЗМІСТ

| | |
|--|----|
| ВСТУП..... | 6 |
| 1. ЗАДАЧА МОНІТОРИНГУ НАФТОВИХ РОЗЛИВІВ НА МОРСЬКІЙ ПОВЕРХНІ МЕТОДОМ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ..... | 8 |
| 2. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ | 10 |
| 2.1 Сімейство продуктів ArcGis | 11 |
| 2.2 Висновки до розділу | 11 |
| 3. ЗАСОБИ РОЗРОБКИ СИСТЕМИ МОНІТОРИНГУ НАФТОВИХ РОЗЛИВІВ НА МОРСЬКІЙ ПОВЕРХНІ МЕТОДОМ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ | 12 |
| 4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ МОНІТОРИНГУ НАФТОВИХ РОЗЛИВІВ НА МОРСЬКІЙ ПОВЕРХНІ МЕТОДОМ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ..... | 28 |
| 4.1 Опис компонентів системи | 28 |
| 4.1.1 Методи доступу до растрових даних | 28 |
| 4.1.2 Реалізація інтерфейсу..... | 29 |
| 4.1.3 Методи застосування фільтрів і реалізація фільтрів | 30 |
| 4.1.4 Реалізація доступу до бази даних та роботи з нею | 32 |
| 4.2 Алгоритм моніторингу нафтових розливів на морській поверхні методом дистанційного зондування землі..... | 35 |
| Висновки до розділу..... | 42 |
| 5. РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ МОНІТОРИНГУ НАФТОВИХ РОЗЛИВІВ НА МОРСЬКІЙ ПОВЕРХНІ МЕТОДОМ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ..... | 43 |
| Висновки до розділу..... | 51 |
| ВИСНОВКИ | 52 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 53 |

ВСТУП

Аварії при освоєнні Світового океану — явище неминуче, як при всякому впровадженні нових конструкцій або технологічних методів. Тому ця проблема буде стояти перед людством завжди. Забруднення морів нафтою завдає шкоди живим ресурсам, здоров'ю людей, перешкоджає морській діяльності, погіршує якість морської води. Отже, на сьогодні потрібно вміти швидко та оперативно визначати обсяги та межі нафтового забруднення, для його подальшого усунення.

Дистанційне зондування Землі уявляє собою незамінний інструмент вивчення та постійного моніторингу нашої планети, що допомагає ефективно використовувати і управляти її ресурсами. Зображення, отримані супутниками дистанційного зондування Землі, знаходять застосування у багатьох галузях — сільському господарстві, геологічних і гідрологічних дослідженнях, і, насамперед, для виявлення, попередження і моніторингу забруднення нашої планети.

Визначення обсягів забруднення нафтою методом дистанційного зондування Землі є дуже важливою та досить актуальною темою на сьогодні. Сучасні матеріали космічних зйомок мають роздільну здатність на місцевості від десятків кілометрів до десятків сантиметрів, отже це є досить потужним джерелом інформації.

Тож, нафтове забруднення є досить важливою проблемою для людства, було вирішено розробити автоматизовану інформаційну систему, яка б надавала можливість обробляти вхідні знімки, надавати інформацію, щодо забруднення, на основі оброблених знімків.

Виявлення нафтових розливів морів відбувається методом обробки, накладання фільтрів на космічний знімок. При цьому слід враховувати умови, за яких було виконано космічну зйомку. Насамперед слід звернути увагу на погоду та рівень шумів на знімку, а також кількість хвиль на воді.

До переваг даної системи можна віднести інтуїтивно зрозумілий інтерфейс, можливість завантажувати у систему будь-який знімок, відносно швидка обробка.

Також, доволі простий алгоритм, що дозволяє розширювати систему, для вирішування більш специфічних задач та випадків.

У даній роботі викладаються теоретичні питання, пов'язані з отриманням даних зі знімків та їх обробки, а також описано процес розробки власне автоматизованої системи для визначення забруднення морів нафтою методом дистанційного зондування Землі.

Для реалізації даного програмного забезпечення було обрано платформу .Net, фреймворки ADO.Net, Entity Framework, ArcObject SDK .Net та мову програмування C#. Розробка інтерфейсу додатку відбувається за допомогою API WinForms. В якості середовища розробки – Microsoft Visual Studio 2017 Community.

У першому розділі пояснювальної записки сформульовано мету й описано постановку задачі.

У другому — розділі наводяться приклади схожих реалізованих систем.

У третьому розділі обґрунтовано вибір програмних засобів.

У четвертому розділі описується програмна реалізація створення інформаційної системи.

У п'ятому розділі подано методику роботи користувача з програмною системою.

1. ЗАДАЧА МОНІТОРИНГУ НАФТОВИХ РОЗЛИВІВ НА МОРСЬКІЙ ПОВЕРХНІ МЕТОДОМ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ

Метою розробки є створення програмного продукту, призначеного для моніторингу нафтових розливів на морській поверхні методом дистанційного зондування Землі, візуалізації результатів, можливості перегляду списку знімків, можливість внесення, видалення та редагування параметрів знімків.

Для досягнення поставленої мети необхідно:

- проаналізувати вже існуючі програмні рішення;
- проаналізувати літературні джерела на відповідну тематику;
- обрати засоби та інструменти розробки програмного продукту;
- обрати методи розробки;
- розробити архітектуру програмного забезпечення;
- здійснити програмну реалізацію системи моніторингу нафтових розливів на морській поверхні методом дистанційного зондування Землі з використанням обраних технологій програмування.

Програмна система має включати наступний функціонал:

- відкриття на налаштування вхідного знімку;
- обробка знімку, для отримання даних про нафтові розливи на морській поверхні;
- збереження результатів обробки;
- візуалізація отриманих даних у вигляді нового зображення, з топоприв'язкою і усіма доступними каналами;
- проведення виявлення з можливістю вибору точності.

Вхідними даними програмної системи мають бути:

- різноманітні зображення, отримані методом дистанційного зондування землі, в тому числі і з застосуванням радарів і радіолокаторів.

Вихідними даними мають бути:

— нове зображення, що візуально демонструє межі та обсяги нафтових розливів на морській поверхні, яке має прив'язку і канали відповідно до вхідного зображення.

Основними користувачами програмної системи мають бути:

— науковці, котрим дана система може бути корисна при проведенні спостережень у водах;

— екологи, котрим дана система може бути корисна для виявлення і оцінки нафтових розливів на морській поверхні.

2. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

Далі пропонується опис програмних рішень для моніторингу нафтових розливів на морській поверхні. Платформа Sentinel Application Platform (SNAP) Desktop

Архітектура SNAP ідеально підходить для обробки та аналізу спостережень Землі завдяки наступним технологічним нововведенням: розширюваність, портативність, модульна платформа багатих клієнтів, загальна абстракція даних EO, керування плитковою пам'яттю та графічна обробка.

Основні характеристики:

- Загальна архітектура для всіх інструментів;
- Дуже швидке відображення зображень і навігація навіть з гіга-піксельних зображень;
- Фреймворк Graph Processing Framework (GPF): для створення визначених користувачем ланцюгів обробки;
- Удосконалене керування шарами дозволяє додавати та маніпулювати новими накладаннями, такими як зображення інших діапазонів;
- Доволі потужний інструмент кластеризації і класифікації;

Для виявлення меж нафтових забруднень морської поверхні використовуючи систему Sentinel Application Platform (SNAP) Desktop слід застосувати алгоритми класифікації, вбудовані у Sentinel Application Platform (SNAP) Desktop.

Головним недоліком цього підходу є низька точність результатів, особливо, коли невідомо, чи присутні на знімку нафтові розливи, чи ні.

2.1 Сімейство продуктів ArcGis

Програмні продукти компанії ESRI (США), найстарішого у світі виробника програмних засобів ГІС (фірма заснована в 1969 році).

Програмні продукти ArcGis порівняно з іншими програмами роботи з географічною інформацією мають ряд переваг:

- строга топологічність даних;
- контроль за цілісністю і топологічністю даних;
- розвинутий апарат роботи з системою координат і географічними проєкціями;
- наявність розвинутого математичного апарату обробки просторових даних.

Основними її недоліками є деякі труднощі при освоєнні редактора, а також заплутана і не завжди зрозуміла ліцензійна політика і велика вартість ліцензій.

2.2 Висновки до розділу

Отже, виходячи з вище описаного в даному розділі, можна зробити висновок, що хоча ArcGis та Sentinel Application Platform (SNAP) Desktop дійсно досить потужні інструменти, для отримання адекватного результату, вони потребують певних навичок роботи з ними. Програма, розроблена в ході виконання дипломної роботи, також вирішує задачу моніторингу нафтових розливів на морській поверхні методом дистанційного зондування землі, проте вона набагато простіша для використання, адже вона не потребує знань і навичок з налаштування і виправлення результатів класифікації та має інтуїтивно зрозумілий інтерфейс користувача.

3. ЗАСОБИ РОЗРОБКИ СИСТЕМИ МОНІТОРИНГУ НАФТОВИХ РОЗЛИВІВ НА МОРСЬКІЙ ПОВЕРХНІ МЕТОДОМ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ

Коли було проаналізовано задачу дипломної роботи, було прийнято рішення розробити додаток до ArcMap. Вести розробку додатку до ArcMap було вирішено мовою C# на платформі .NET з використанням ArcObject SDK 10.4 та Entity Framework. Розробка велася у інтегрованому середовищі розробки Microsoft Visual Studio 2015 Enterprise.

Мова C# – одна з об'єктно-орієнтовних мов програмування високого рівня, що була розроблена для платформи Microsoft .NET. Ця мова має строгу статичну типізацію, хоча нещодавно було додано можливість використовувати динамічну типізацію. Мова C# є дуже потужним інструментом розробки програмного забезпечення, також ця мова є дуже популярною, на рівні з Java. Головними перевагами даної мови є дуже швидкий розвиток цієї мови, це пов'язано з її популярністю та затребуваністю. З кожною новою версією додається новий функціонал та нові синтаксичні конструкції, які призначені спрощувати та прискорювати майбутню розробку.

Мову C# можна вважати одною з найбільших новинок у галузі створення мов програмування. На відміну середини 20-го століття, коли активно створювалися мови програмування, на даний момент мови програмування створюються доволі не часто, адже зараз вже є доволі багато потужних мов програмування і всі ресурси витрачаються на їх вдосконалення та оновлення. Тому, нова мова повинна мати в собі найкращі якості вже існуючих мов програмування, щоб отримати визнання та популярність.

Мову C# підтримують найбільш популярні операційні системи, такі як: Linux, багато модифікації Unix (такі, як HP-UX, Solaris і OpenBSD), Windows, Mac OS X, RISC OS, та багатьох інших. Особливо визначною подією, був вихід кроссплатформеного .NET Core.

Серед основних переваг мови C# можна назвати такі [7, 8]:

- чистий синтаксис (для виділення блоків слід використовувати відступи);
- стандартний дистрибутив має велику кількість корисних модулів;
- наявність у вільному доступі дуже потужного інтегрованого середовища розробки(IDE) Microsoft Visual Studio;
- Сі подібний синтаксис мови, дуже зручно для розробників, знайомих з мовами, наприклад, C++ або Java;
- уніфікована система типізації.
- C# створювався паралельно з каркасом Framework .Net і повною мірою враховує всі його можливості - як FCL, так й CLR;
- C# є повністю об'єктно-орієнтованою мовою, де навіть типи, вбудовані в мову, представлені класами;
- C# є потужною об'єктною мовою з можливостями спадкування й універсалізації;
- C# є спадкоємцем мов C/C++, зберігаючи кращі риси цих популярних мов програмування;
- завдяки платформі .NET Framework, що стала надбудовою над операційною системою, розробники C# одержують ті ж переваги роботи з віртуальною машиною, що й програмісти Java. Ефективність коду навіть підвищується, оскільки виконавче середовище CLR являє собою компілятор проміжної мови, у той час як віртуальна Java-машина є інтерпретатором байт-коду;
- потужна бібліотека каркасів додатків дозволяє легко та швидко будувати Веб-служби та інші компоненти, також це досить спрощує роботу з базами даних;
- реалізація, що сполучає побудову надійного й ефективного коду, є немаловажним чинником, що сприяє успіху C#.

Платформа Microsoft .NET Framework – програмна технологія, що була запропонована компанією Microsoft як платформа для створення як звичайного десктопного програмного забезпечення, так і веб-додатків .

Платформа .NET Framework - це технологія, яка підтримує розробку і запуск нового покоління застосунків і веб-служб XML. При розробці платформи .NET Framework враховувалися наступні цілі:

- забезпечення збереження принципів об'єктно-орієнтованого програмування для локального збереження і виконання коду, для локального виконання коду, розподіленого в Інтернеті, або для віддаленого виконання;
- забезпечення середовища виконання коду, що зменшує вірогідність конфліктів при розгортанні програмного забезпечення та управлінні версіями;
- забезпечення середовища виконання коду, що гарантує безпечне виконання коду, включаючи код, створений невідомим або не повністю довіреною стороннього постачальника;
- забезпечення середовища виконання коду, що виключає проблеми з продуктивністю середовищ виконання сценаріїв або інтерпретується коду;
- забезпечення єдиних принципів розробки для різних типів додатків, таких як додатки Windows і веб-додатки;

Платформа .NET Framework складається з загальномовного середовища виконання [7, 8] (середовища CLR) і бібліотеки класів .NET Framework. Основою платформи .NET Framework є середовище CLR. Середовище виконання можна вважати елементом, який керує кодом під час виконання і надає основні служби, такі як методи управління пам'яттю, управління потоками і віддалена взаємодія. При цьому, середовище виконання накладає свої умови якості та валідності коду, що забезпечують надійність та безпеку готового коду. Як відомо, основною задачею середовища виконання є управління кодом до і під час виконання. Загалом, код можна умовно поділити на два види: керований і некерований. Керованим кодом називається код, що звертається до середовища виконання, натомість некерований код – не звертається до середовища виконання.

Бібліотекою класів називають загальну колекцію класів і інтерфейсів (тобто типів), що застосовуються в розробці програмного забезпечення.

Платформа .NET Framework може розміщуватися некерованими компонентами, які завантажують середу CLR у власні процеси і запускають

виконання керованого коду, створюючи таким чином програмне середовище, що дозволяє використовувати кошти як керованого, так і некерованого виконання[10]. Ця платформа може не тільки надавати набір базових середовищ виконання, але також може підтримувати створення базових середовищ виконання незалежними виробниками програмного забезпечення.

Наприклад, при розробці програмного забезпечення на фреймворку ASP.NET, середовище виконання і масштабоване середовище для керованого коду розміщується на серверній стороні, при чому сам ASP.NET працює безпосередньо з середовищем виконання, для того щоб була змога виконувати застосунки ASP.NET.

Браузер Internet Explorer можна вважати прикладом некерованого додатку, який має своє середовище виконання (у вигляді розширень типів MIME). Розміщення середовища виконання в браузері Internet Explorer дозволяє додавати керовані елементи або елементи управління WinForms та WPF в HTML-документи. Таке розміщення середовища дозволяє запускати керований мобільний код і користуватися його істотними перевагами, зокрема виконанням в умовах неповної довіри і з ізольованим зберіганням файлів.

На наступному малюнку демонструється взаємозв'язок середовища CLR і бібліотеки класів з одними додатками і всією системою. На рисунку (Рисунок 3.1) також показано, як керований код працює в межах ширшої архітектури.

Керований код — це термін, що було запропоновано корпорацією Microsoft для визначення вихідного коду комп'ютерної програми, який, для свого виконання віртуальну машину, причому виконується керований код виключно віртуальною машиною[8].

Слово керований тут належить до способу обміну даними між програмою і середовищем виконання. Воно означає, що в будь-якій точці виконання, виконавче середовище може припинити виконання й отримати дані, специфічні для поточного стану[7].

Необхідні для цього дані надані в керованому коді у мові Intermediate Language та в пов'язаних з цим кодом метаданих.

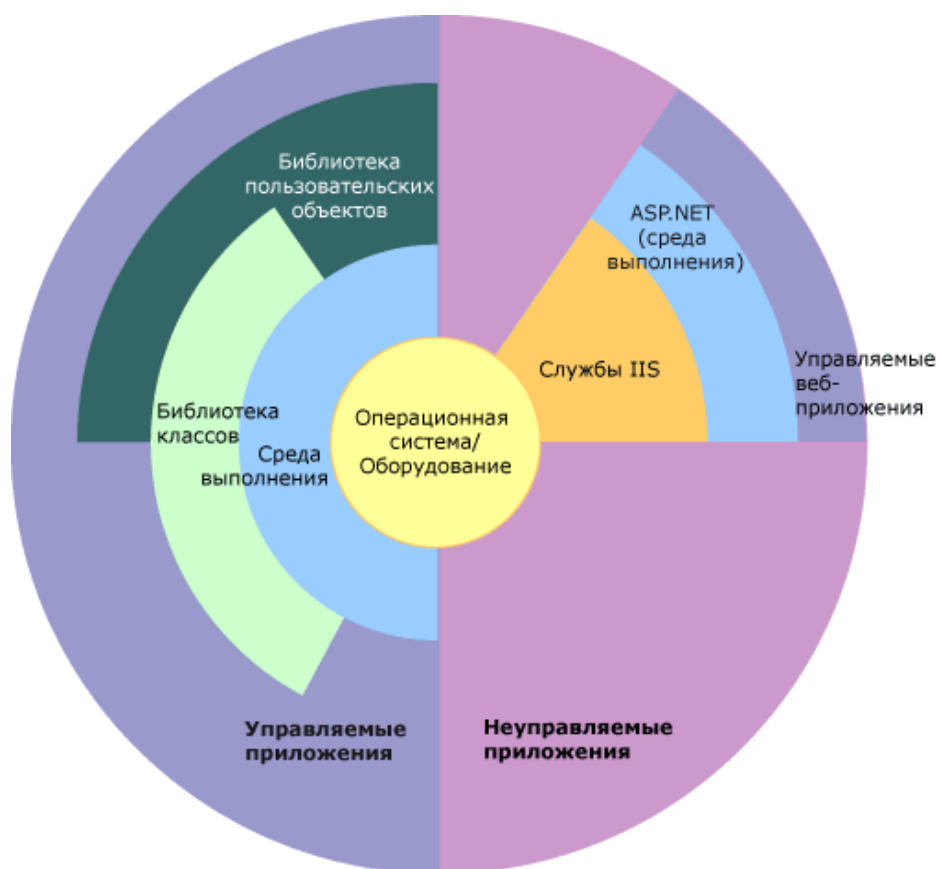


Рисунок 3.1 – Работа керованого коду

Взаємодія на основі промислових стандартів, яка гарантує інтеграцію коду платформи .NET Framework з будь-яким іншим кодом.

Однією з ідей .NET є сумісність служб, написаних різними мовами. Саме ця платформа дає такі можливості як:

- доступ і робота з базами даних;
- робота з мовами запитів;
- створення користувацького інтерфейсу інтерфейсу;
- створення веб-додатків;
- підтримка різних мов програмування.

Бібліотека класів платформи .NET Framework являє собою колекцію типів, які тісно інтегруються із середовищем CLR [11, 12]. Бібліотека класів є об'єктно-

орієнтованої. Вона надає типи, від яких керований код користувача може успадковувати функції. Це не тільки спрощує роботу з типами даних .NET Framework, але і скорочує час вивчення нових засобів платформи .NET Framework. Крім того, компоненти незалежних розробників можна поєднувати з класами платформи .NET Framework.

Наприклад, в класах колекцій .NET Framework описується і реалізується набір інтерфейсів для розробки призначених для користувача класів колекцій[12]. На рисунку (Рисунок 3.2) також показано основні види колекцій .NET Framework. Призначені для користувача класи колекцій легко об'єднуються з класами та інтерфейсами .NET Framework.



Рисунок 3.2 – Основні види колекцій

Призначені для користувача класи колекцій легко об'єднуються з класами та інтерфейсами .NET Framework.

Типи, що входять до бібліотеки .NET Framework дозволяють вирішувати поширені завдання програмування, включаючи роботу з векторами, збір даних, підключення і робота з базами даних, а також доступ до файлів. На кінець, до цих звичайним завданням бібліотека класів містить типи, які підтримують спеціалізовані сценарії розробки. Платформу .NET Framework (рисунок 3.3) використовується для розробки наступних типів додатків і служб[11]:

- консольні додатки;

Архитектура Microsoft .NET Framework

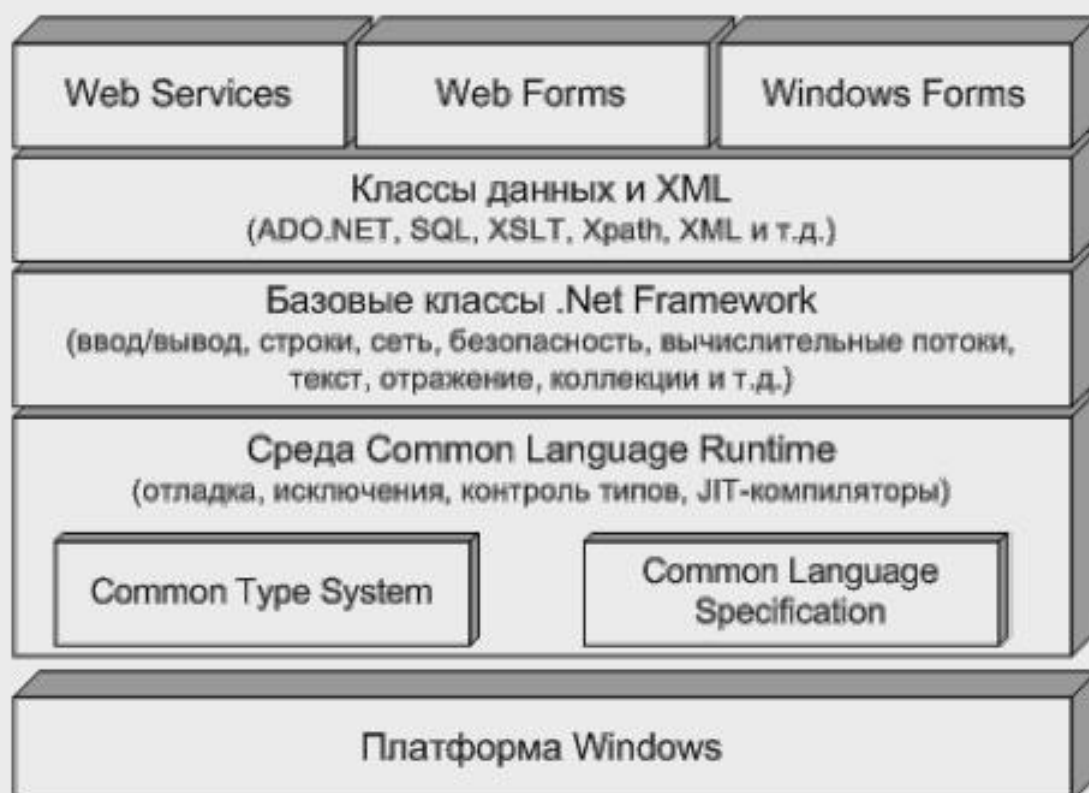


Рисунок 3.3 – Архітектура .NET Framework

- додатки з графічним інтерфейсом Windows (Windows Forms);

- додатки Windows Presentation Foundation (WPF);
- програми ASP.NET;
- служби Windows;
- сервісно-орієнтовні програми для використання Windows Communication Foundation (WCF);
- додатки, що підтримують бізнес-процеси Windows Workflow Foundation (WF).

Середовище розробки Microsoft Visual Studio – це лінійка програмних продуктів від фірми Microsoft, які включають в себе інтегроване середовище розробки програмного забезпечення (IDE) та ряд інструментів для розробки програмного забезпечення. Середовище розробки Microsoft Visual Studio дозволяє створювати як спеціалізовані програмні додатки, так і програми з інтерфейсом користувача, в тому числі з підтримкою технології Windows Forms, WPF, а також сайти, веб-додатки, веб-служби як в нативному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. Середовище розробки Visual Studio - доволі потужний програмний продукт, використання для розробки програмних продуктів якого продовжується вже десятки років.

Середовище розробки Microsoft Visual Studio включає в себе редактор коду з підтримкою технології IntelliSense і можливістю базового рефакторингу коду. Вбудований відлагоджувач може працювати як відлагоджувач коду, так і відлагоджувач машинного рівня. Інші вбудовані інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу, веб-редактор, дизайнер схеми бази даних і дизайнер класів. Середовище розробки Microsoft Visual Studio дозволяє створювати і підключати сторонні додатки для розширення функціональності практично на кожному рівні, включаючи підтримку системи контролю версій. Базовий інструмент для роботи з додатками, NuGet, зображено на рисунку 3.4.

Ідеї Visual Studio дуже сильно пов'язані з ідеєю .NET, дозволяють вважати її принципово новою розробкою, що визначає новий етап у створенні програмних продуктів. Виділю дві найважливіші, на мій погляд, ідеї:

- відкритість для мов програмування;
- принципово новий підхід до побудови архітектури.

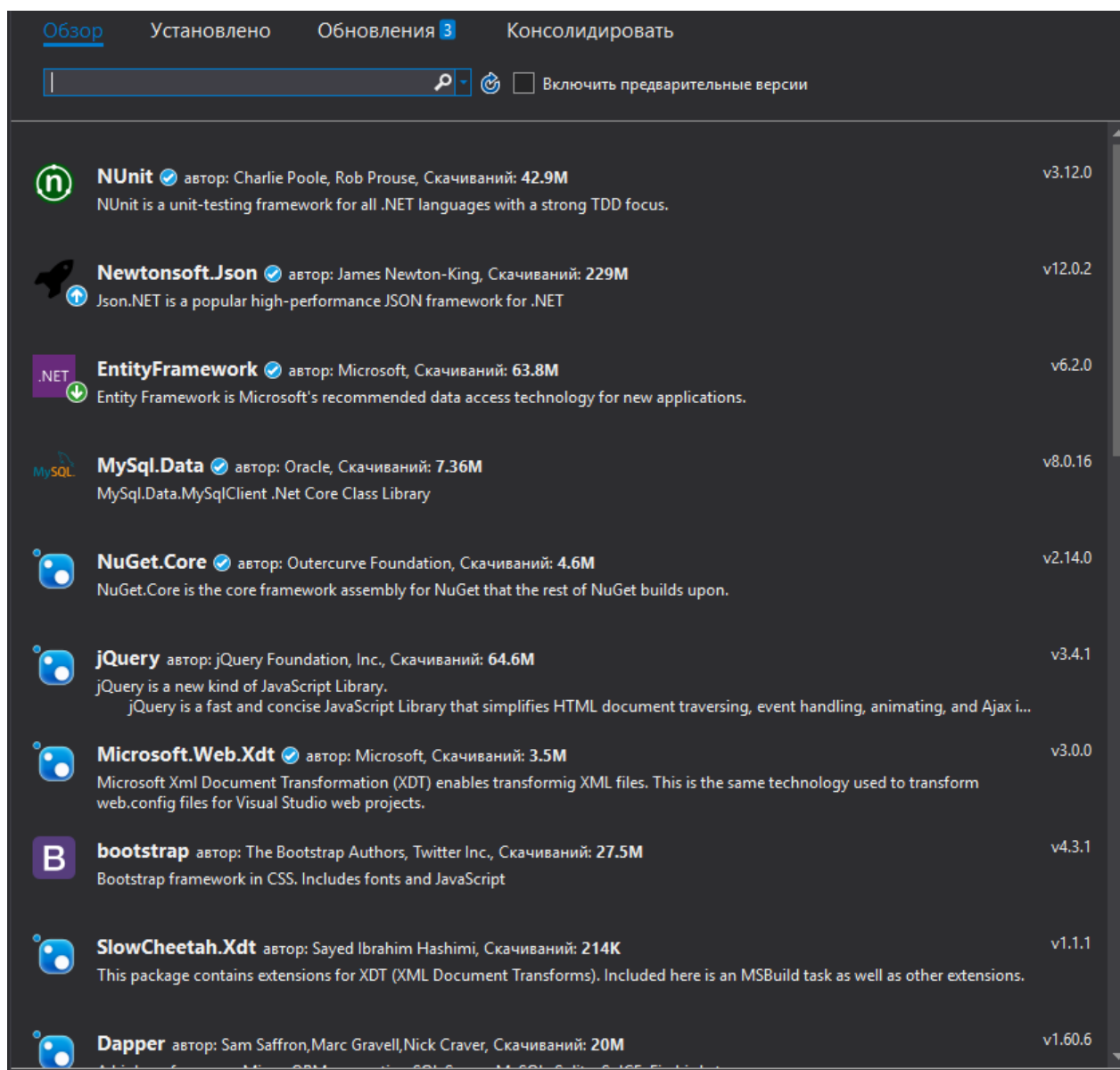


Рисунок 3.4 – диспетчер пакетів NuGet

Програмний продукт, розроблений у ході дипломної роботи включає в себе роботу з бд. Базою даних для додатку було обрано MS SQL Server. Для створення і

адміністрування бази даних було використано внутрішній редактор баз даних Microsoft Visual Studio 2015 та утиліту Microsoft SQL Server Manager Studio.

Microsoft SQL Server [12] – комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства.

Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву Transact-SQL (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями[13]. T-SQL дозволяє використовувати додатковий синтаксис збережених процедур і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком). Microsoft SQL Server та Sybase ASE для взаємодії з мережею використовують протокол рівня застосунка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних).

Сервер Microsoft SQL Server також підтримує Open Database Connectivity (ODBC) — інтерфейс взаємодії застосунків з системою управління базами даних. Версія SQL Server 2005 надає можливість підключення користувачів за допомогою веб-сервер-сервіси, що використовують протокол SOAP. Цей підхід дає змогу програмам, що не пристосовані до Windows з'єднуватись і працювати з MS SQL Server. Microsoft також випустила сертифікований драйвер JDBC, що дозволяє застосункам під керування Java (таким як BEA і IBM Websphere) з'єднуватись з Microsoft SQL Server 2000 і 2005.

SQL Server підтримує дзеркалювання та кластеризацію баз даних [13]. Кластер серверу SQL—це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.

SQL Server 2005 також має вбудовану підтримку .NET Framework. Завдяки цьому підходу, збережені процедури, можуть бути написані на будь-якій мові платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework. На відміну від інших процесів в системі, фреймворк .NET Framework виділяє додаткову пам'ять і будує інструменти керування SQL Server, не використовуючи вбудовані засоби Windows. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server.

Сервер MS SQL Server — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

Сервер MS SQL Server вважається гарним сервером баз даних для малих, середніх і великих систем. MS SQL Server підтримується на великій кількості платформ. Найповніше можливості сервера виявляються в UNIX-системах, таких як Linux, Ubuntu, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MS SQL Server:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

Серед основних переваг MS SQL Server відзначають наступні:

- можливість горизонтально масштабувати MS SQL Server, отже це допомагає підтримувати досить великі бази даних;
- MS SQL Server є кроссплатформеним;
- зв'язаність. MS SQL Server має мережеву структуру. До MS SQL Server можна одержувати доступ із будь-якої точки Internet кільком користувачам одночасно. MS SQL Server має цілий ряд програмних інтерфейсів додатків (Application Programming Interface –API), які дозволяють встановлювати з'єднання з

MS SQL Server із додатків, написаних на таких мовах як C, C++, C#, Perl, PHP, Java, Python;

- MS SQL Server має систему контролю доступу до даних, забезпечує шифрування даних при передаванні;
- швидкість функціонування;
- зручність експлуатації. MS SQL Server досить зручно встановлюється та реалізується, легко адмініструється.

Утиліта SQL Server Management Studio (Рисунок 3.5) – утиліта компанії Microsoft для Microsoft SQL Server для конфігурування, керування та адміністрування усіх компонентів Microsoft SQL Server.

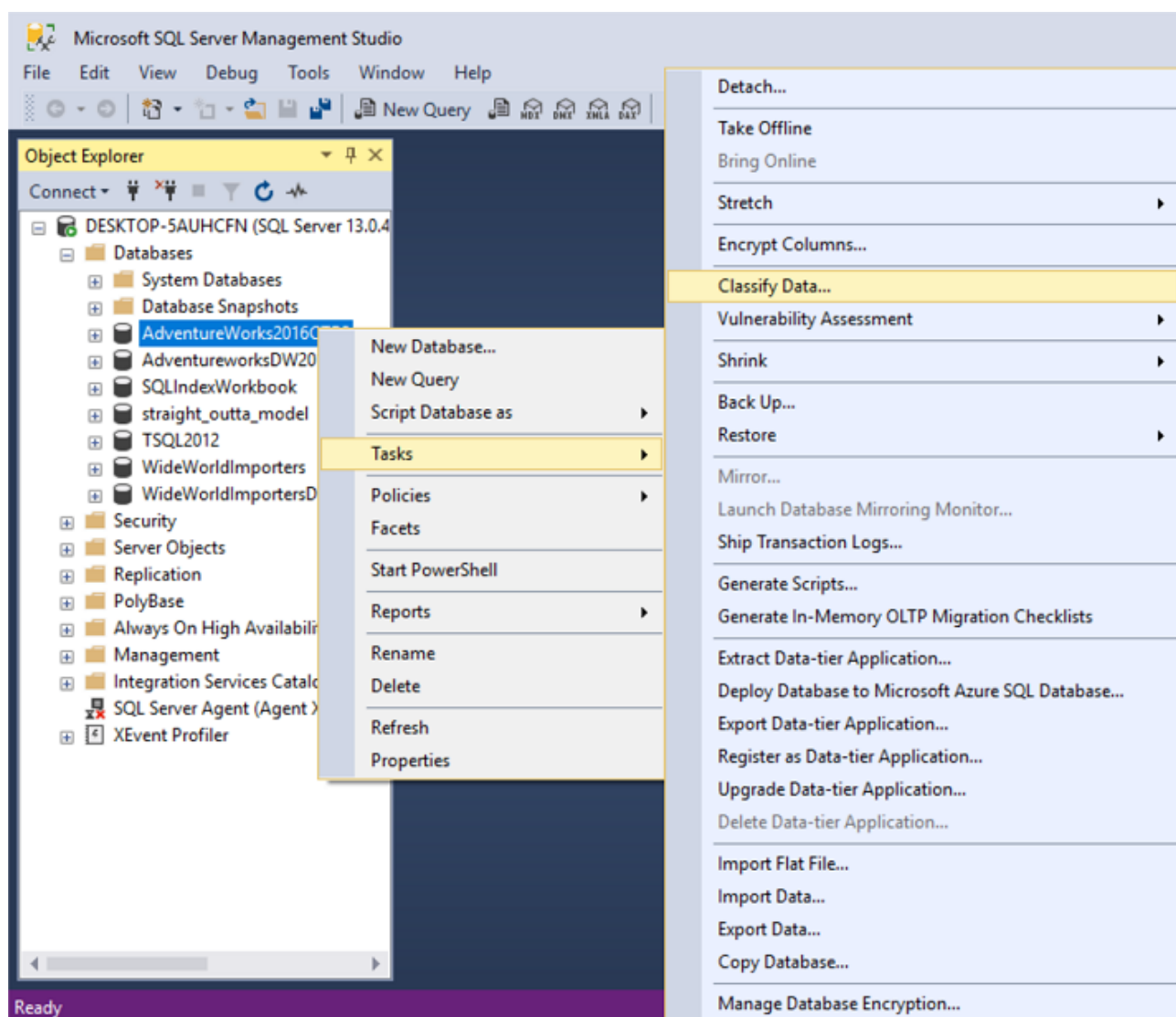


Рисунок 3.5 – інтерфейс SQL Server Management Studio

Утилита має в собі редактор скриптів та графічну програму, котра працює з об'єктами та налаштуваннями серверу. Головним інструментом SQL Server Management Studio є Object Explorer, який дозволяє користувачу проглядати, витягувати об'єкти сервера, а також повністю керувати ними.

Розроблена програма створена у вигляді додатку до ArcMap, для розробки було використано ArcObjects SDK .NET.

Бібліотека ArcObjects – це бібліотека компонентних компонентних компонентів, що складають основу ArcGIS. Для роботи з ArcObjects в середовищі розробки .NET створено набір основних збірок (PIA) і бібліотек об'єктів COM для керування взаємодією .NET-COM. Компоненти ArcObjects встановлюються з продуктами ArcGIS Desktop, ArcGIS Engine або ArcGIS Server і можуть використовуватися для виконання наступних задач:

- налаштування програми ArcGIS Desktop;
- для побудови окремих програм відображення;
- для розробки веб-додатків;

Цей комплект розробки програмного забезпечення (SDK) для розробки за допомогою ArcObjects у .NET допоможе розробникам у написанні своїх додатків шляхом надання кращих практик, концептуальної документації, розділів коду та зразків.

Інструмент розробки ArcObjects SDK .NET поєднує вміст, що раніше містився в ArcGIS Desktop і ArcGIS Engine. До складу ArcObjects SDK для .NET входять документація, зразки, засоби розробки та шаблони Visual Studio для розробки ArcObjects. Вона зосереджена на наступних трьох основних категоріях програм ArcObjects:

- Створення надбудов для ArcGIS Desktop;
- Створення автономних додатків;
- Розробка за допомогою ArcGIS Server.

Нова система створення додатків до ArcGIS надає розробникам декларативно створену основу для створення блоків користувацьких функціональних можливостей в межах одного стисненого файлу. Ці файли-додатки можуть бути

легко розподілені між користувачами, не покладаючись на програми встановлення або реєстрацію об'єктної моделі компонента (COM). Файли додатків можна встановити, скопіювавши їх у відоме.

На рисунку 3.9 показано опцію менеджера надбудов і діалогове вікно в інтерфейсі користувача ArcMap.

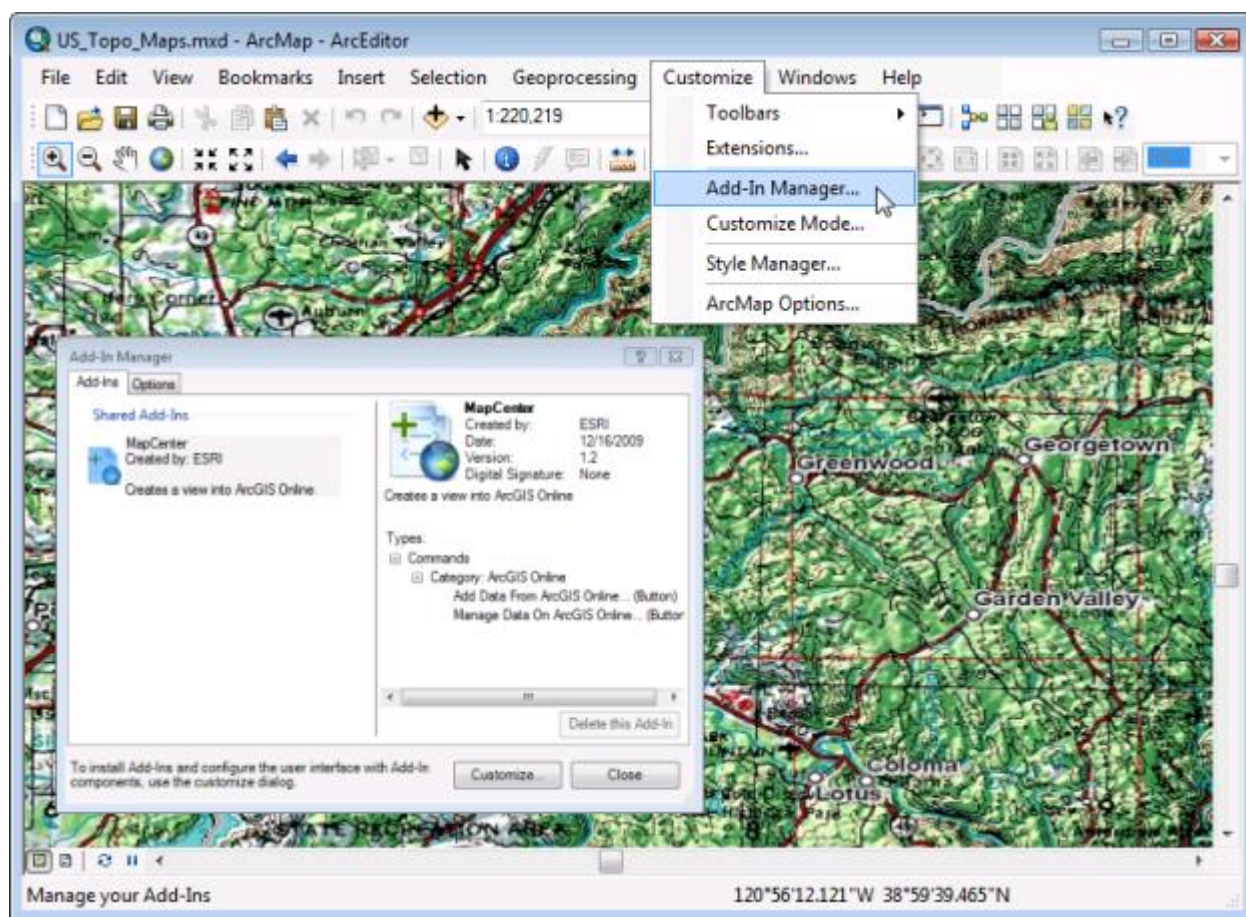


Рисунок 3.6 – менеджер додатків в інтерфейсі користувача ArcMap

Файли надбудови можна також розділяти між користувачами в організації за допомогою спільного доступу до мережі.

Додатки до ArcGIS можна створювати за допомогою VB .NET, C # або Java. Щоб спростити розробку, ESRI надає додаткові майстри, які інтегруються з популярними середовищами розробки, включаючи Eclipse і Microsoft Visual Studio. Користувачі можуть отримувати інформацію про джерело, вміст та інші деталі через метадані XML (Extensible Markup Language). Ця інформація може бути використана

для виявлення додатків і надбудов у веб-порталах, фільтрації або обмеження надбудов на основі політик у споживчій організації та загалом керування додатками і надбудовами, коли вони копіюються на машині користувача.

Додатки підтримують наступні типи додаткових компонентів:

- Команди та інструменти – це прості елементи Microsoft Windows Forms, які можна додавати на панелях інструментів. У випадку з кнопками, кнопки можна додавати на меню;
- Комбо-бокси – це текстове поле, містить розкривний список елементів і може додатково забезпечувати область редагування;
- Меню та контекстні меню – меню містить випадаючий список кнопок, підменю та інших підтримуваних елементів. Пункти меню можуть підключатися з вбудованих джерел, додаткових джерел, або їх комбінації. Меню, як правило, розміщують на панелі інструментів, але їх можна використовувати незалежно, як контекстні (спливаючі) меню та кореневі меню;
- Панелі інструментів – може містити у собі всі інші додаткові елементи, такі як кнопки, палітри інструментів і меню. Як і в меню, додаткові елементи, які знаходяться на панелі інструментів можуть підключатися вбудованих джерел, додаткових джерел, або їх комбінації;
- Палітри інструментів – забезпечують компактний спосіб групування відповідного набору інструментів. Останній використаний інструмент з'являється на панелі інструментів поруч із невеликою кнопкою, що випадає, для доступу до інших інструментів у групі. Як і в меню, додаткові елементи, які знаходяться на панелі інструментів можуть підключатися вбудованих джерел, додаткових джерел, або їх комбінації;
- Прикріплені вікна – це плаваючі або закріплені вікна, які відображаються в програмах ArcGis Desktop. Дані вікна можуть бути заповненими будь-яким вмістом: діаграми, слайд-шоу, міні-карти або користувацькі діалогові вікна, що містять інші елементи керування, включаючи елементи ESRI. Також є можливість задавати початкове положення даних вікон та можливість налаштування групування;

– Розширення додатків і редакторів – дозволяють налаштовувати робочі процеси редагування, підключаючи їх безпосередньо до рамки редагування.

Висновки до розділу

Отже, виходячи з інформації, наданої у цьому розділі, можна зробити висновок, що обрані засоби повністю підходять для вирішення поставленої задачі.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ МОНІТОРИНГУ НАФТОВИХ РОЗЛИВІВ НА МОРСЬКІЙ ПОВЕРХНІ МЕТОДОМ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ

Програмна реалізація даної системи представляє собою три основні взаємодіючих компоненти:

- середовище ArcMap;
- методи доступу до растрових даних;
- методи застосування фільтрів до растрових даних і реалізація самих фільтрів.

Програму було реалізовано за допомогою платформи .Net з використанням API WinForms та написано на мові програмування C#.

4.1 Опис компонентів системи

Розроблену систему умовно можна розділити на три основні частини:

- методи доступу до растрових даних;
- методи застосування фільтрів до растрових даних і реалізація самих фільтрів;
- інтерфейс користувача;
- реалізація роботи з базою даних.

Методи доступу до растрових даних забезпечують можливість обробки набору растрових даних, відкритих у середовищі ArcMap. Методи застосування фільтрів до растрових даних і реалізація самих фільтрів містять у собі сам алгоритм вирішення задачі і його застосування до растрових даних.

4.1.1 Методи доступу до растрових даних

Методи доступу до растрових даних, які забезпечують роботу з растровими даними шару було розроблено за допомогою .NET Framework та ArcObject SDK.

В наборі бібліотек класів ArcObject SDK є ряд специфічних класів, інтерфейсів і методів для роботи з растровими даними, у тому числі: створення нового набору растрових даних, доповнення набору, редагування набору, видалення.

Основна суть методів доступу до растрових даних полягає у наступних пунктах:

- отримання растрового шару з відкритого документу у ArcMap. Далі, отриманий растровий шар можна буде розглядати під інтерфейсом растра. За замовчуванням отримується нульовий шар документу;
- налаштування робочого простору, в якому буде створюватись новий растровий шар;
- створення нового растрового шару. Новий шар створюється на основі растра, отриманого раніше. Висота, ширина, канали та топоприв'язка нового растрового шару задається у повній відповідності до початкового растрового шару.
- реалізація роботи з блоком пікселів. На цьому етапі реалізовується безпосередній доступ до пікселів растрового шару. Створюються блоки пікселів та отримується доступ до піксельних матриць кожного каналу.

4.1.2 Реалізація інтерфейсу

Інтерфейс користувача реалізовано за допомогою API WinForms та ArcObject SDK.

Технологія Windows Forms – це технологія створення додатків під Windows (API – application programming interface), що відповідає за графічний інтерфейс та є частиною платформи .NET. Даний інтерфейс дуже сильно спрощує доступ до елементів інтерфейсу Microsoft Windows за рахунок створення обгортки для існуючого Win32 API в керованому коді. Застосунок Windows Forms представляє собою застосунок, який є подієорієнтовним. Тобто, на відміну від стандартних пакетних програм, більша частина часу витрачається на очікування від

користувача якої-небудь дії, наприклад, заповнення текстового поля або ж кліку мишкою по кнопці.

Інструмент розробки ArcObjects SDK дозволяє створювати та розміщувати елементи Windows Forms в самому ArcMap, на панелі інструментів. У даному випадку ці елементи створюються у вигляді додатку до ArcMap (ArcMap Add-in), створення такого додатку зображено на рисунку 4.1.

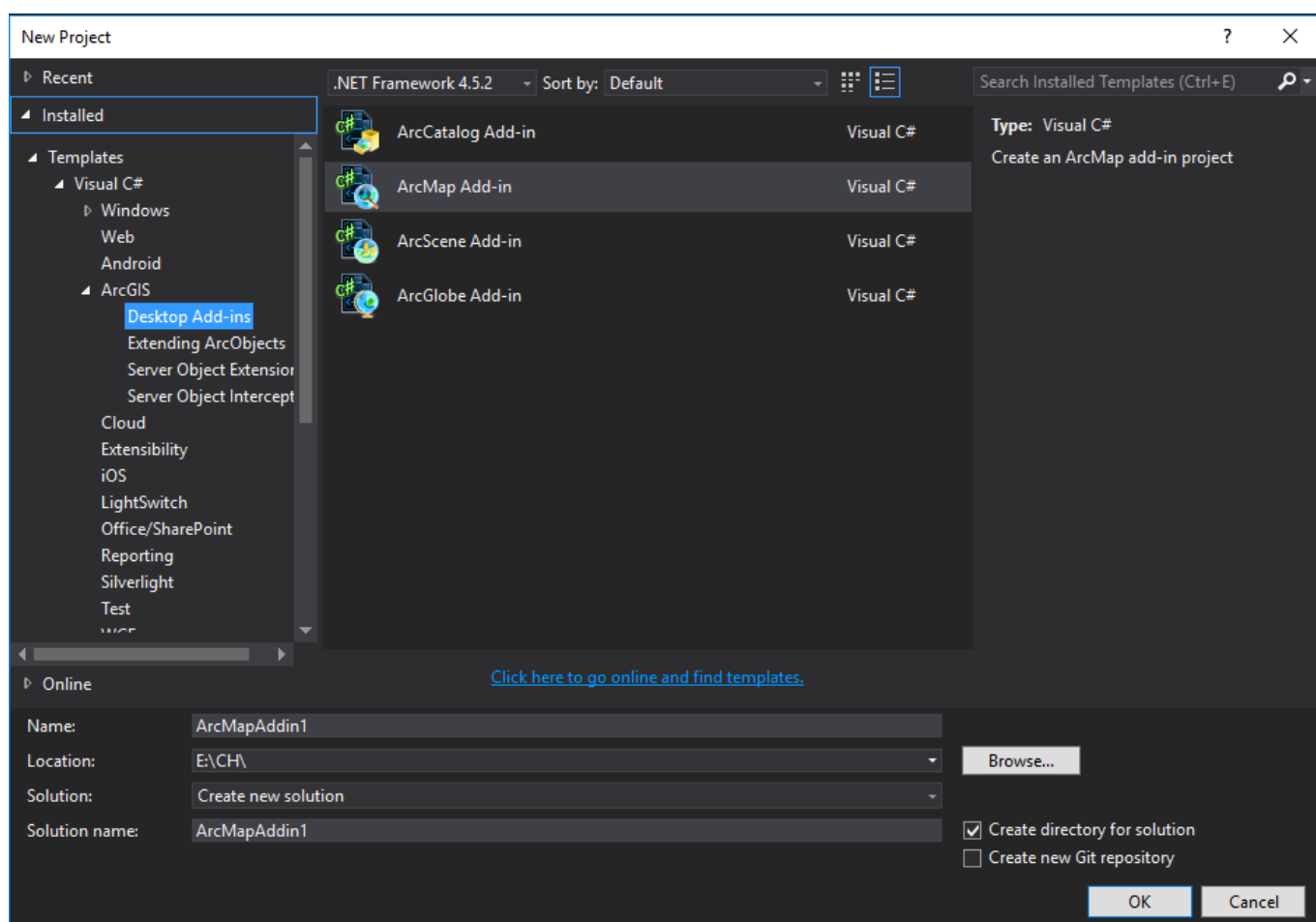


Рисунок 4.1 – створення додатку за допомогою ArcObjects SDK

Налаштувати зовнішній вигляд майбутнього додатку можна як при його створенні (рисунок 4.2), так і після, використовуючи меню налаштування.

4.1.3 Методи застосування фільтрів і реалізація фільтрів

Загалом, застосування методи застосування фільтрів мають у своїй основі роботу з набором матриць, як мінімум з матрицею початкового растрового шару і матрицею результуючого растрового шару.

Для зручної роботи з матрицями було обрано .NET Collection Framework та LINQ.

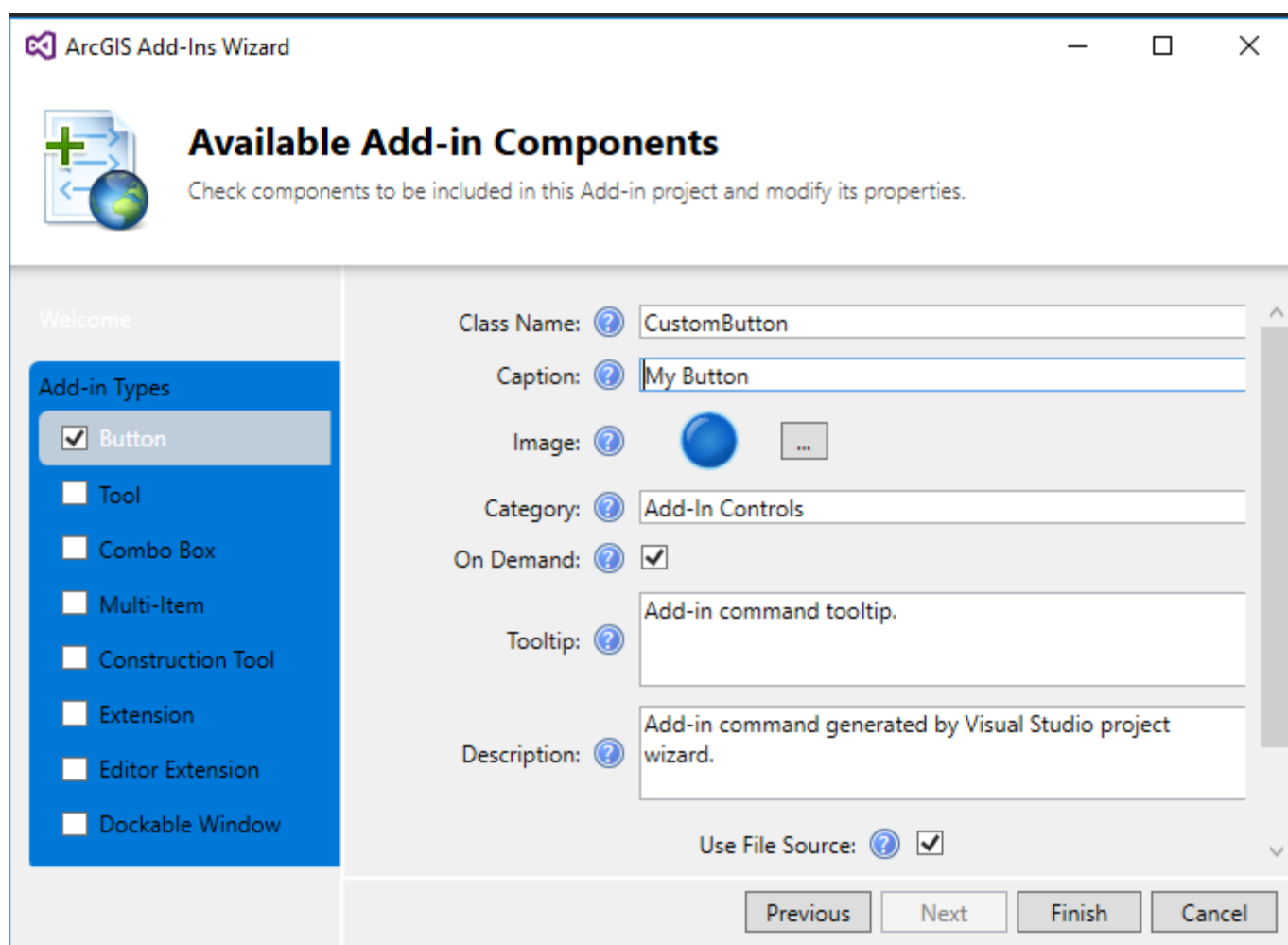


Рисунок 4.2 – налаштування зовнішнього вигляду додатку до ArcMap

Фреймворк .NET Collection Framework – досить потужний і зручний інструмент роботи з наборами даних. Робота з масивами не завжди є зручною, особливо у випадку, коли на момент компіляції ми не знаємо розміри масивів, які потрібно буде обробляти. І в такому випадку набагато зручніше використовувати

.NET Collection Framework, структури даних якого не мають даних обмежень. Ще одна перевага колекцій над масивами полягає у тому, що більшість з них реалізують стандартні структури даних, наприклад стек, черга, словник, список, які можуть стати корисними для вирішення специфічних задач.

Основна частина класів колекцій міститься в просторі імен System.Collections та System.Collections.Generic (не узагальнені і узагальнені колекції).

У даній системі було використано двовимірний список для зберігання даних растру і словник для зберігання фільтрів.

4.1.4 Реалізація доступу до бази даних та роботи з нею

Доступ до бази даних реалізується за допомогою зручного фреймворку ADO.NET та Entity Framework.

Фреймворк ADO.NET уявляє собою технологію роботи з даними, яка заснована на платформі .NET Framework[14]. Ця технологія являє нам набір класів, через які ми можливо відправляти запити до баз даних, встановлювати підключення, отримувати відповідь від бази даних і виробляти ряд інших операцій.

Причому важливо зазначити, що систем управління баз даних може бути безліч. У своїй сутності вони можуть відрізнятися. MS SQL Server, наприклад, для створення запитів використовує мову T-SQL, а MySQL і Oracle застосовують мову PL-SQL. Різні системи баз даних можуть мати різні типи даних. Також можуть відрізнятися якісь інші моменти. Однак функціонал ADO.NET побудований таким чином, щоб надати розробникам уніфікований інтерфейс для роботи з самими різними СУБД.

Основу інтерфейсу взаємодії з базами даних в ADO.NET представляє обмежений список об'єктів:

- Підключення – Connection;
- команда – Command,
- дата рідер – DataReader,

- набір даних – DataSet
- адаптер даних – DataAdapter.

За допомогою об'єкта Connection відбувається установка підключення до джерела даних. Об'єкт Command дозволяє виконувати операції з даними з БД. Об'єкт DataReader зчитує отримані в результаті запиту дані. Об'єкт DataSet призначений для зберігання даних з БД і дозволяє працювати з ними незалежно від БД. І об'єкт DataAdapter є посередником між DataSet і джерелом даних. Головним чином, через ці об'єкти і буде йти робота з базою даних.

Функціонально класи ADO.NET можна розбити на два рівня: підключений і відключений (рисунок 4.3). Кожний провайдер даних .NET реалізує свої версії об'єктів Connection, Command, DataReader, DataAdapter і ряду інших, який складають підключений рівень. Тобто за допомогою них встановлюється підключення до БД і виконується з нею взаємодію.



Рисунок 4.3 – схема архітектури ADO .NET

В архітектурі ADO.Net існує прив'язка до фізичної структури даних, тому при написанні коду для звернення до бази даних необхідно пам'ятати схеми таблиць, відношень. Для спрощення написання коду та його автоматичної підтримки компанією Microsoft було розроблено Entity Framework [15], котрий

виводить абстракцію на новий рівень об'єктної моделі. Entity Framework являє спеціальну об'єктно-орієнтовану технологію на базі фреймворка .NET для роботи з даними. Якщо традиційні засоби ADO.NET дозволяють створювати підключення, команди та інші об'єкти для взаємодії з базами даних, то Entity Framework являє собою більш високий рівень абстракції, який дозволяє абстрагуватися від самої бази даних і працювати з даними незалежно від типу сховища. Якщо на фізичному рівні ми оперуємо таблицями, індексами, первинними і зовнішніми ключами, але на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами. Також Entity Framework надає можливість взаємодіяти з об'єктами з використанням як Entity SQL, так і використовуючи LINQ у вигляді LINQ to Entities.

LINQ – проект Microsoft по додаванню синтаксису мови запитів, що нагадує SQL, в мови програмування платформи .Net Framework. LINQ дозволяє використовувати SQL-подібний синтаксис безпосередньо в коді програми, написаної, наприклад на мові C#, використовуючи деякі особливості мови:

- Анонімна типізація
- Методи розширення
- Лямбда-числення
- Дерево виразів

Стандартні оператори мови запитів по типам джерел даних LINQ можна умовно поділити на:

- LINQ до колекцій
- LINQ до об'єктів
- LINQ до баз даних

Від самого початку підтримуючи механізм запитів для колекцій об'єктів у пам'яті, реляційних баз даних та даних у форматі XML, архітектура LINQ може бути розширена, що дозволяє стороннім розробникам реалізовувати доступ до їх джерел даних через механізми LINQ. Для цього необхідно реалізувати стандартні оператори запитів, використовуючи методи розширення, або реалізувати інтерфейс

IQueryable, котрий дозволяє розбирати дерево виразів під час виконання, трансліюючи його у свою мову запитів.

4.2 Алгоритм моніторингу нафтових розливів на морській поверхні методом дистанційного зондування землі

Нафта, потрапляючи у море, утворює на морській поверхні шар плівки різної товщини, залежно від виду нафтового продукту, витік якого стався[1]. За відсутності хвиль і вітру, один кубічний сантиметр нафтового продукту утворює пляму радіусом близько 50 метрів за 1-2 години.

Ці плівки мають специфічні фізичні властивості. По мірі розтікання нафти і зменшення товщини її шару колір плями змінюється від чорного або темно-коричневого, характерного для плям великої товщини, до срібlistого або веселкового, притаманного тонкій плівці на кінцях плями. Напівтверді або в'язкі види нафтопродуктів, замість того, щоб поширюватись у вигляді тонких шарів, розпадаються на фрагменти, які віддаляються один від одного і можуть досягати товщини декілька сантиметрів. З цього можна зробити висновок, що визначати межі нафтових плям за допомогою звичайних знімків доволі складно і не завжди можливо, адже в залежності від виду нафтопродукту, колір плями може дуже сильно варіюватись.

Але, всі нафтові плями також мають одну загальну і важливу властивість. На місті плями майже ніколи не утворюється хвиль, тобто нафтова пляма майже завжди є гладкою з мінімальною кількістю шороховатостей.

Саме ця властивість буде використовуватись у даному алгоритмі. Ефективним засобом моніторингу нафтових розливів на морській поверхні є радіолокаційне зондування, яке здатне реєструвати варіації поверхневого хвилювання моря (рисунок 4.3). Оскільки нафтова пляма утворює плівку на поверхні моря, яка зменшує хвилі, то це можна визначити, обробивши радіолокаційний знімок. В найкращому випадку, навіть без ретельної обробки

можна буде побачити нафтові плями у вигляді чорних плям на фоні незабрудненої поверхні моря, яка буде виглядати як шум.

Але на рівні хвилювання моря можуть впливати різного роду чинники, що можуть сильно ускладнити визначення нафтового забруднення морської поверхні.

Тому при аналізі радіолокаційних знімків з метою визначення меж нафтових розливів слід брати до уваги наступну інформацію:

- період року і імовірність утворення криги на поверхні;
- швидкість вітру, адже при швидкості вітру від 10 метрів в секунду, нафтова пляма, як правило, змивається з морської поверхні.
- форму темної плями на знімку, адже нафтові плями зазвичай мають чіткий край;
- розмір ймовірного нафтового розливу;
- географічне розташування, тобто відстань до берега і наявність островів чи природних перешкод.

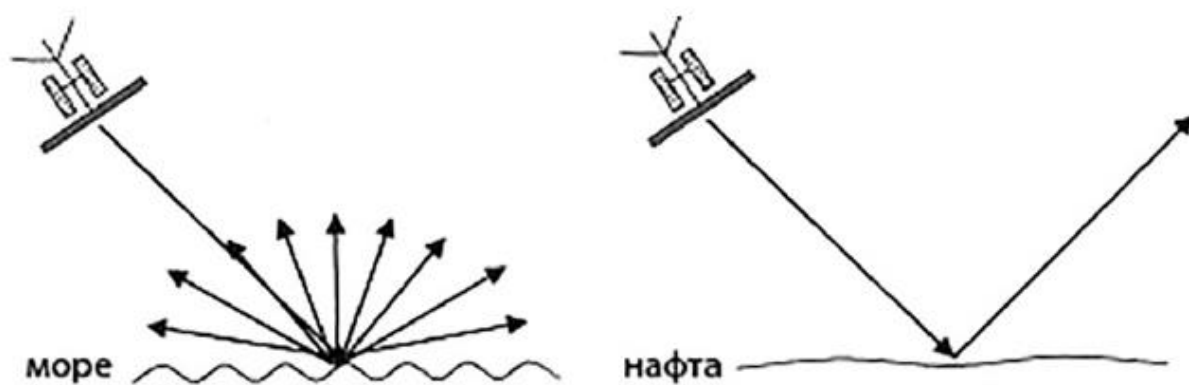


Рисунок 4.3 – виявлення нафтових розливів за допомогою радіолокаційного зондування

На рисунках 4.4 та 4.5 зображені радіолокаційні знімки, на яких зафіксовані нафтові плями.

Загалом, такий підхід доволі ефективний, але у нього є недоліки. Здебільшого, радіолокаційні знімки дуже сильно зашумлені, тому не завжди можна чітко визначити межі нафтових розливів на морській поверхні.

Для прибрання зайвого шуму та збільшення чіткості нафтових плям і було розроблено наступний алгоритм.

Даний алгоритм полягає у поетапному застосуванні ряду фільтрів і методів зменшення шуму, розмиття і збільшення чіткості. Насамперед був застосований алгоритм розмивання Гауса.

Даний ефект широко використовується в графічних програмах, як правило, для зменшення зашумленості зображенні та зниження деталізації. Візуальний ефект цієї фільтрації розмивання аналогічний погляду на зображення крізь напівпрозорий екран, суттєво відрізняючись від ефекту боке, який можна отримати за допомогою несфокусованого об'єктива або тіні об'єкта при звичайному освітленні.



Рисунок 4.4 – нафтові плями на радіолокаційному знімку

Цей тип фільтру використовує функцію Гауса для перерахунку значення кожного пікселя, відносно сусідніх пікселів. Коли цей метод застосовується у двох вимірах, в результаті отримується поверхня, контури якої є околами Гаусівського

розподілу з центрального пікселя. Значення цього розподілу використовується для формування матричного фільтру, таблиця 4.1. Цей підхід надає ефект розмивання зображення, яке доволі добре зберігає кордони та краї, що необхідно при визначенні меж розливів нафтопродуктів.

Сила розмивання залежить від розміру матриці розмивання. Оптимальним варіантом є використання матриць 5 на 5 та 8 на 8.

Проте, оцінивши складність алгоритму роботи з матрицею, що за нотацією O становить $O(h * w * n * n)$, де h і w – ширина і висота зображення, n – розмірність фільтру, можна зробити наступну оптимізацію:



Рисунок 4.5 – нафтова пляма на радіолокаційному знімку

– виконувати горизонтальне і вертикальне розмиття окремо, що дозволить добитися складності в $2 * O(h * w * n)$.

На таблиці 4.1 можна побачити фільтр для розвивання Гауса розмірністю 8.

Таблиця 4.1 – матриця фільтру для розмивання Гауса

| | | | | | | |
|-------------------|------------|-------------------|-------------------|-------------------|------------|-------------------|
| 0.00000067 | 0.00002292 | 0.00019117 | 0.00038771 | 0.00019117 | 0.00002292 | 0.00000067 |
| 0.00002292 | 0.00078633 | 0.00655965 | 0.01330373 | 0.00655965 | 0.00078633 | 0.00002292 |
| 0.00019117 | 0.00655965 | 0.05472157 | 0.11098164 | 0.05472157 | 0.00655965 | 0.00019117 |
| 0.00038771 | 0.01330373 | 0.11098164 | 0.22508352 | 0.11098164 | 0.01330373 | 0.00038771 |
| 0.00019117 | 0.00655965 | 0.05472157 | 0.11098164 | 0.05472157 | 0.00655965 | 0.00019117 |
| 0.00002292 | 0.00078633 | 0.00655965 | 0.01330373 | 0.00655965 | 0.00078633 | 0.00002292 |
| 0.00000067 | 0.00002292 | 0.00019117 | 0.00038771 | 0.00019117 | 0.00002292 | 0.00000067 |

Також у ході роботи застосовуються наступні фільтри, порівняння яких показано на таблиці 4.2:

– Медіанний фільтр – його робота заключається в тому, що кожен піксель, замінюється значенням медіани у вікні фільтра, рисунок 4.6.

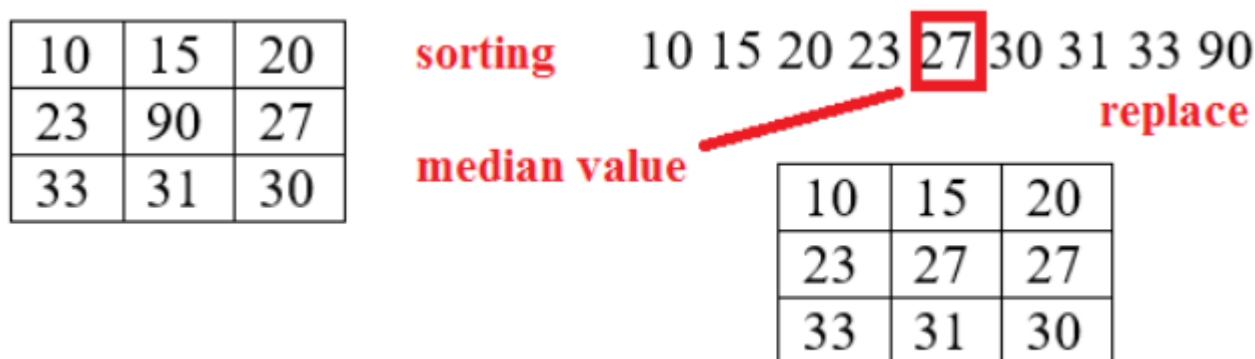


Рисунок 4.6 – накладання медіанного фільтру з рамкою 3 на 3

Цей фільтр є корисним інструментом для видалення імпульсного шуму (наприклад, спекл) – шуми менше половини вікна фільтра можуть бути придушені або навіть видалені, не розмиті, як у випадку середнього фільтра. Медіанний фільтр дозволяє зберегти край, хоча це може призвести до видалення (або придушення) малих (також лінійних) об'єктів з зображення, точно так само, як і видалення (або придушення) спекл-шуму.

– Фільтр середнього арифметичного – його робота заключається у тому, що кожен піксель замінюється середнім значенням у вікні фільтра. Вона не видаляє

спекл з зображення, а усереднює його в одне. Теоретично, темні і яскраві спекл-пікселі у вікні фільтра можуть вимикати один одного. Імовірність таких ситуацій зростає з величиною вікна фільтра. Проте він створює розмиття зображення, втрату деталей і, зрештою, втрату роздільної здатності.

– Локальний регіон (Local region) – він розділяє вікно фільтра на вісім регіонів і обчислює дисперсію для кожного з них. Потім кожен піксель замінюється на середнє значення найбільш гомогенної області (тобто з найменшою дисперсією) - припускається, що вона менше впливає на спекл-шум.

– Фрост-фільтр – цей фільтр виведений з алгоритму середньої квадратичної помилки. Її ядро адаптується до локальних статистичних значень зображення, тому може пригнічувати спекл[4].

– Лі фільтр – це перший модельний фільтр, присвячений придушенню спекл-шуму. Він також заснований на алгоритмі середньої квадратичної похибки.

Таблиця 4.2 – порівняння результатів фільтрації з використанням різних фільтрів

| Тип фільтру | Розмір рамки | Середнє значення | Стандартне відхилення |
|----------------------------|--------------|------------------|-----------------------|
| Фільтр середнього значення | 1 | 8.4 | 8.6 |
| | 2 | 6.7 | 5.7 |
| | 3 | 5.3 | 4.3 |
| | 4 | 4.4 | 3.4 |
| | 5 | 3.7 | 2.9 |
| Медіанний фільтр | 1 | 7.1 | 9.0 |
| | 2 | 4.3 | 5.9 |
| | 3 | 2.6 | 3.8 |
| | 4 | 1.6 | 2.7 |
| | 5 | 1.1 | 2.0 |

| | | | |
|---------------------------|---|-----|------|
| Фільтр локальних регіонів | 1 | 8.2 | 10.8 |
| | 2 | 6.3 | 8.4 |
| | 3 | 5.1 | 6.0 |
| | 4 | 4.6 | 4.8 |
| | 5 | 4.4 | 4.1 |
| Фрост фільтр | 1 | 8.6 | 9.1 |
| | 2 | 7.1 | 6.6 |
| | 3 | 5.9 | 5.0 |
| | 4 | 5.1 | 3.9 |
| | 5 | 4.7 | 3.2 |
| Фільтр Лі | 1 | 8.6 | 8.8 |
| | 2 | 7.1 | 6.5 |
| | 3 | 5.9 | 5.6 |
| | 4 | 5.1 | 5.4 |
| | 5 | 4.7 | 5.5 |
| Гамма-фільтр | 1 | 8.6 | 8.6 |
| | 2 | 7.0 | 5.9 |
| | 3 | 5.6 | 4.4 |
| | 4 | 4.7 | 3.6 |
| | 5 | 4.0 | 3.0 |
| Фільтр Гаусса | 1 | 8.4 | 8.6 |
| | 2 | 6.8 | 5.9 |
| | 3 | 5.4 | 4.5 |
| | 4 | 4.5 | 3.7 |
| | 5 | 3.8 | 3.2 |

— Гамма-фільтр — ґрунтується на припущенні, що початкове значення погіршеного пікселя лежить між місцевим середнім та його фактичним значенням.

Висновки до розділу

Отже, розроблену систему умовно можна розділити на чотири основні частини:

- методи доступу до растрових даних;
- методи застосування фільтрів до растрових даних і реалізація самих фільтрів;
- інтерфейс користувача;
- реалізація роботи з базою даних.

Методи доступу до растрових даних забезпечують роботу з растровими даними шару було розроблено за допомогою .NET Framework та ArcObject SDK. Методи застосування фільтрів до растрових даних і реалізація самих фільтрів є основою даної роботи, у цій частині прописані алгоритми обробки і накладення фільтрів для отримання очікуваного результату. Ці фільтри зберігаються у базі даних, тому для роботи з ними реалізована робота з базою даних, використовуючи технологію Entity Framework.

5. РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ МОНІТОРИНГУ НАФТОВИХ РОЗЛИВІВ НА МОРСЬКІЙ ПОВЕРХНІ МЕТОДОМ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ

Для встановлення розробленої програмної системи персональний комп'ютер має відповідати наступним апаратним вимогам (Таблиця 5.1) і вимогам з програмного забезпечення (таблиця 5.2):

Таблиця 5.1 – Апаратні вимоги

| | |
|----------------------------|--|
| Частота процесора | Не менше 2.2 ГГц |
| Платформа | x86 або x64 з розширенням SSE2 |
| Пам'ять/ОЗП | Мінімум: 4 Гб Рекомендована: 8 Гб |
| Властивості відображення | Глибина кольору 24 розряди |
| Роздільна здатність екрану | Не менше 1024x768 |
| Відео/графічний адаптер | Мінімум 256 Мб відеопам'яті Графічний прискорювач з підтримкою 24-розрядного кольору Необхідно OpenGL версії 2.0 runtime або вище, рекомендовано Shader Model 3.0 або вище |

Таблиця 5.2 – Вимоги до програмного забезпечення

| | |
|------------------------------------|--|
| Вимоги до Microsoft .NET Framework | Версія Microsoft .NET Framework 4.6 або більш нова |
|------------------------------------|--|

| | |
|--------|------------------------------|
| ArcGis | ArcGis desktop версії 10.4.1 |
|--------|------------------------------|

Щоб інсталиувати розроблену програмну систему, потрібно виконати наступні кроки:

- інсталиувати ArcGis версії 10.4.1;
- встановити .NET Framework 4.6.*, або перевірити, чи від його було встановлено;
- встановити файл *.ESRIAddIn за допомогою утиліти ESRIRegasm.

Для запуску системи, в першу чергу необхідно відкрити у ArcMap проект з потрібним знімком (рисунок 5.1):

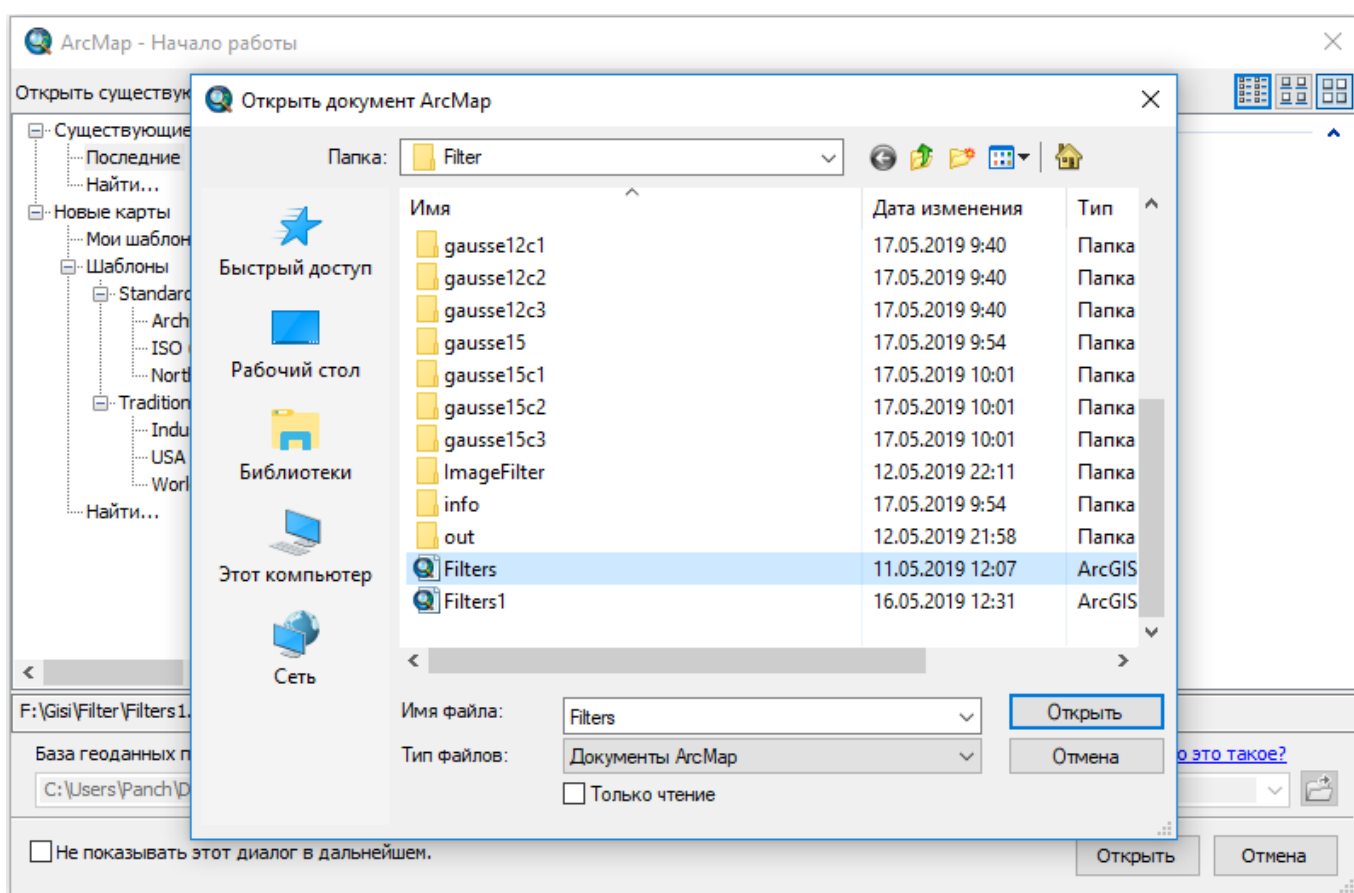


Рисунок 5.1 – відкриття документу в ArcMap

Для того, щоб отримати доступ до інструменту для визначення нафтового

забруднення, потрібно вивести його на панель інструментів. Його можна додати до будь-якої панелі, але краще створити свою.

Для створення користувацької палені інструментів потрібно зайти в налаштування панелей (рисунок 5.2), для цього достатньо натиснути праву кнопку миші і обрати пункт налаштування.

Налаштування панелей є досить потужним інструментом з налаштування та конфігурування зовнішнього вигляду проекту в ArcMap.

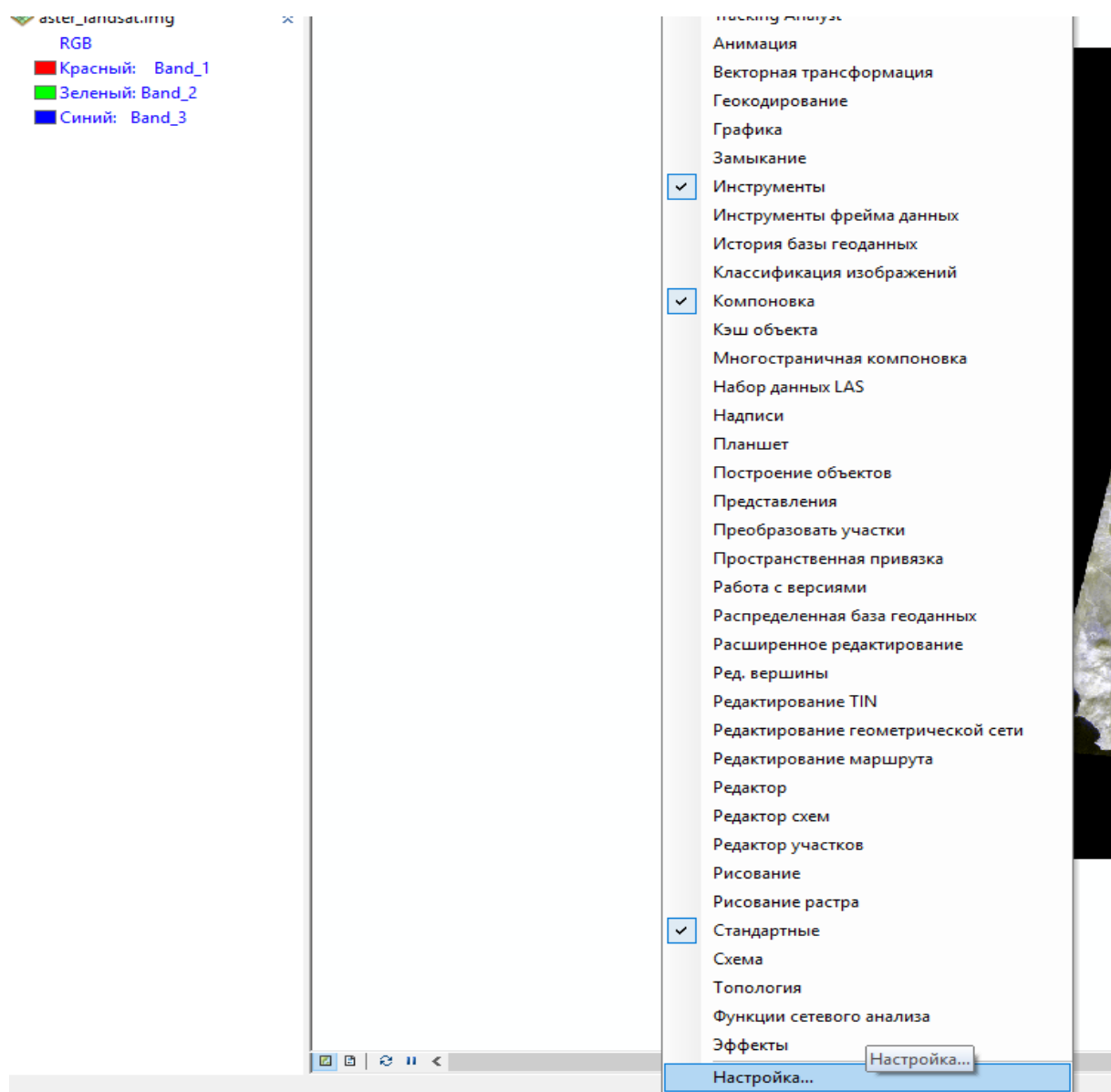


Рисунок 5.2 – меню налаштування панелей

Після цього, потрібно створити нову панель через меню. Для цього в меню

обрати необхідно натиснути кнопку новий (рисунок 5.3) та вказати назву нової панелі інструментів.

У цьому меню можна, також, налагоджувати та налаштовувати вже існуючі панелі та додатки.

Панелі можна створювати та конфігурувати у вкладці панелі інструментів. Додатки та базові інструменти можна налаштовувати у вкладці команди.

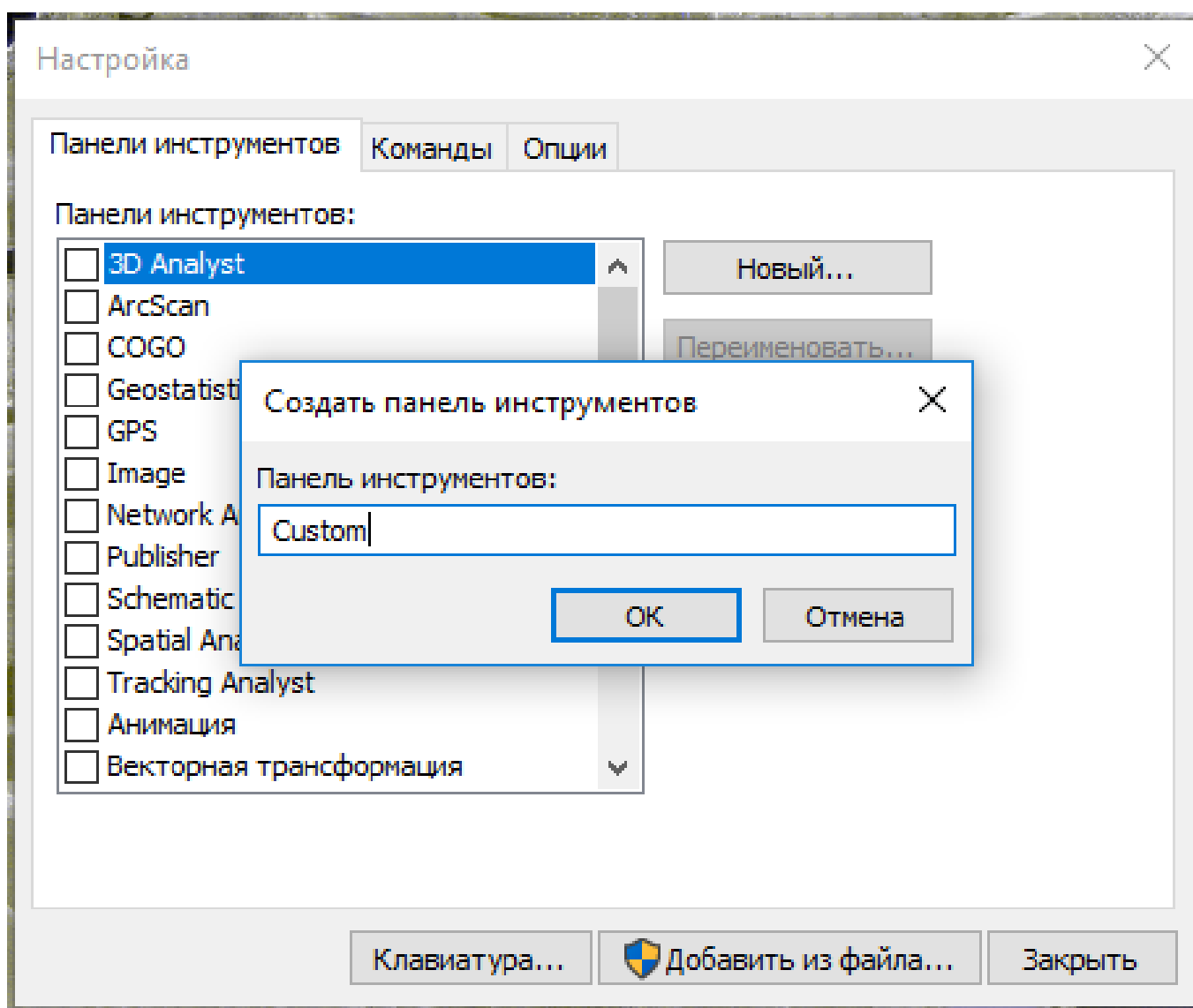


Рисунок 5.3 – створення панелі інструментів

Після того, як нова панель була створена, потрібно додати на неї потрібні інструменти. Для цього слід зайти в налаштування цієї панелі (рисунок 5.4) і у

розділі команди обрати потрібний елемент і перетягнути його на панель.

Потрібний елемент можна знайти у вкладці “команди”, за категорією “Add-In Controls”, де знаходяться всі додатки до ArcMap.

Потрібний додаток називається “Filter”. Тепер його треба перетягнути на потрібну панель інструментів. Після перетягування, додаток буде додано на панель і ним можна буде користуватися.

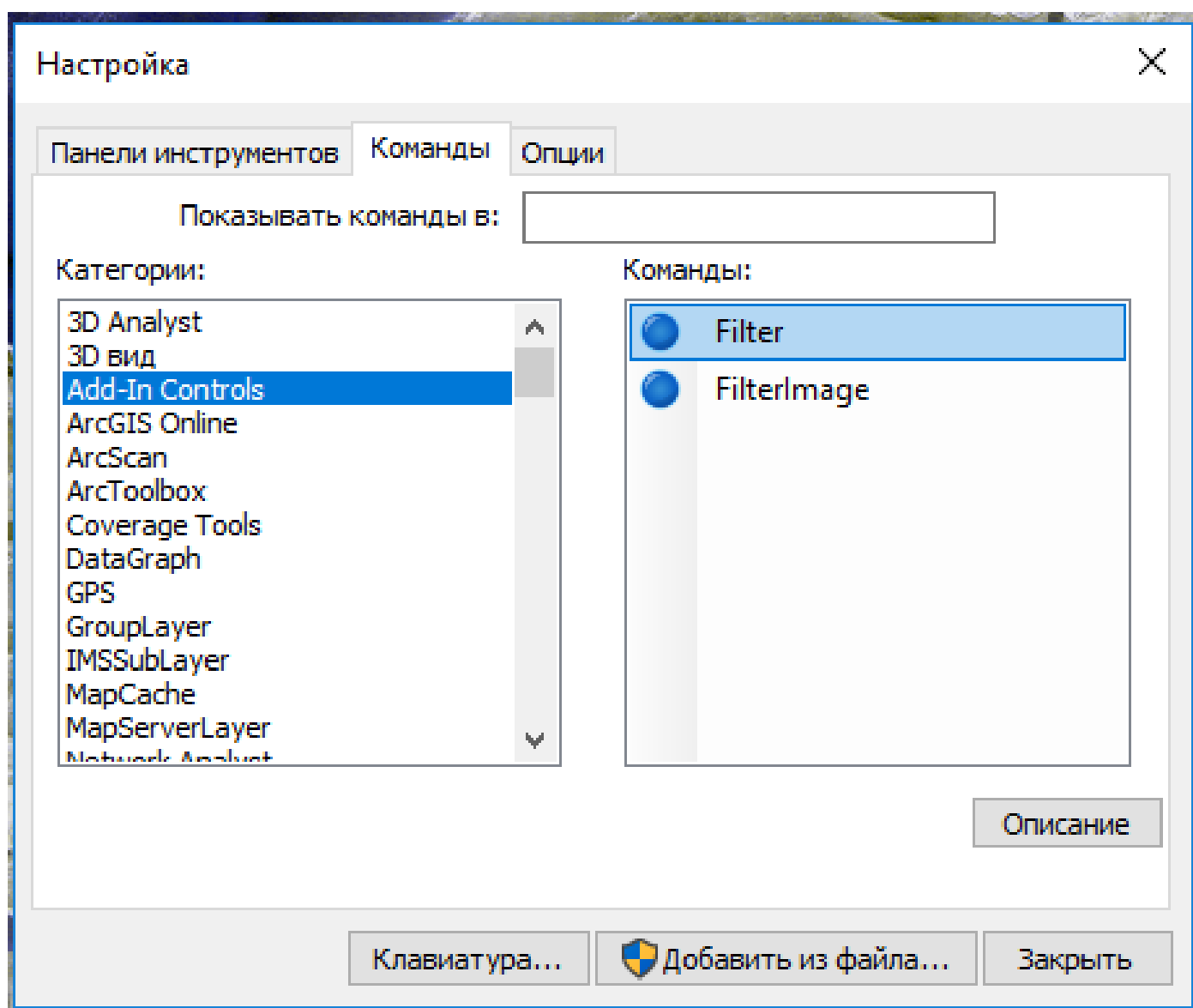


Рисунок 5.4 – вікно додання і видалення додатків

Тепер інструмент додано на панель і можна налаштувати його зовнішній вигляд (рисунок 5.5).

Можна налаштувати:

- назву елементу;
- стиль відображення (тільки текст, тільки значок, значок і текст);
- можна вибрати зображення для значка елементу.

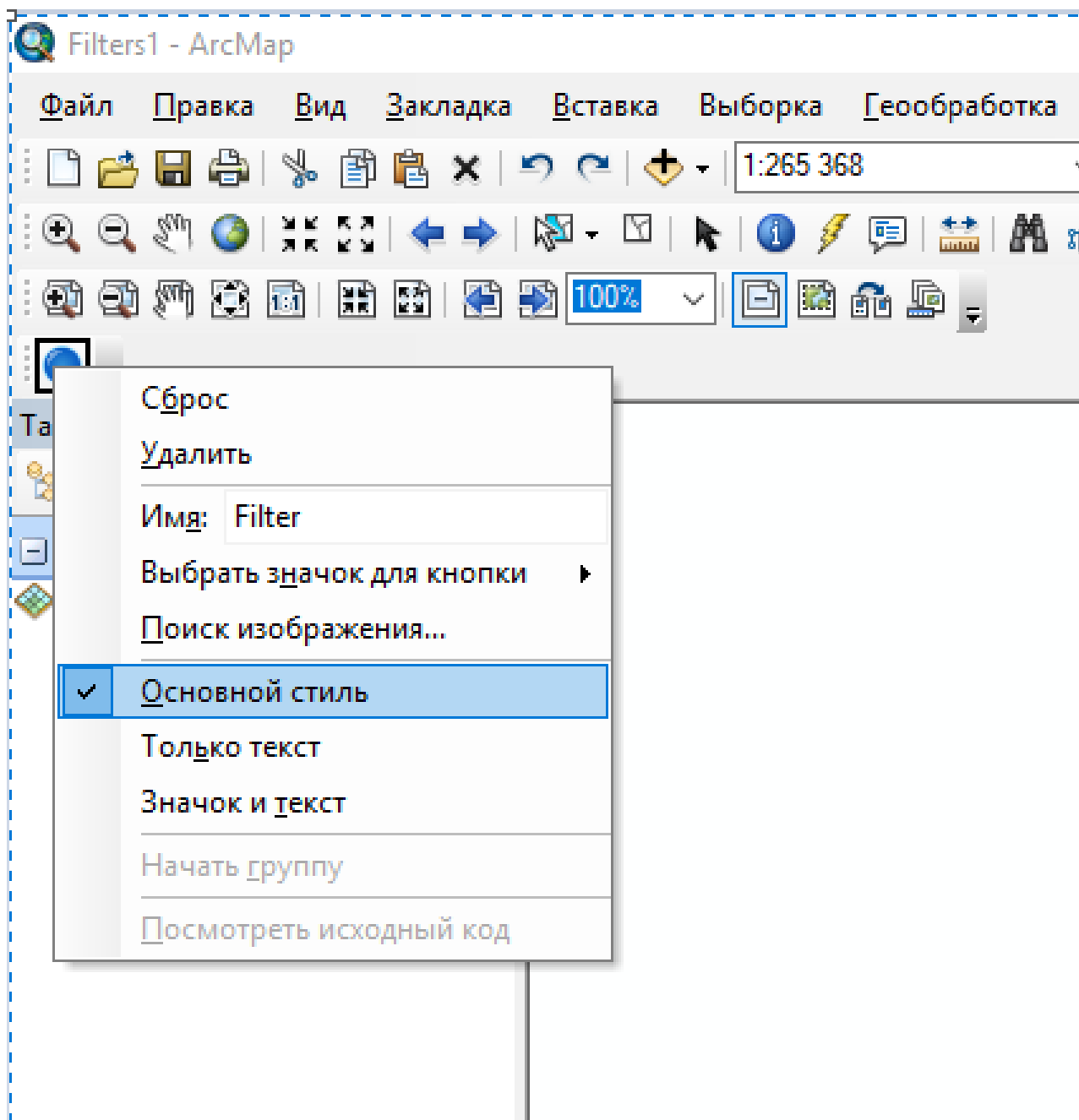


Рисунок 5.5 – налаштування зовнішнього вигляду користувацького елементу

Після всіх налаштувань інструмент готовий до роботи. Цей інструмент представлений у вигляді кнопки, і для запуску додатку достатньо натиснути цю

кнопку.

При натисненні на кнопку, на знімок, відкритий у даному проєкті ArcMap застосовується ряд фільтрів. Сама операція накладання фільтрів алгоритмічно не складна, але на знімках з великою роздільною здатністю та на просто великих знімках, ця операція буде виконуватись доволі довго.

Для обробки зображення на 16 мільйонів пікселів потрібно приблизно 5-7 годин роботи програми.

На рисунку 5.6 показано частину знімку до обробки.

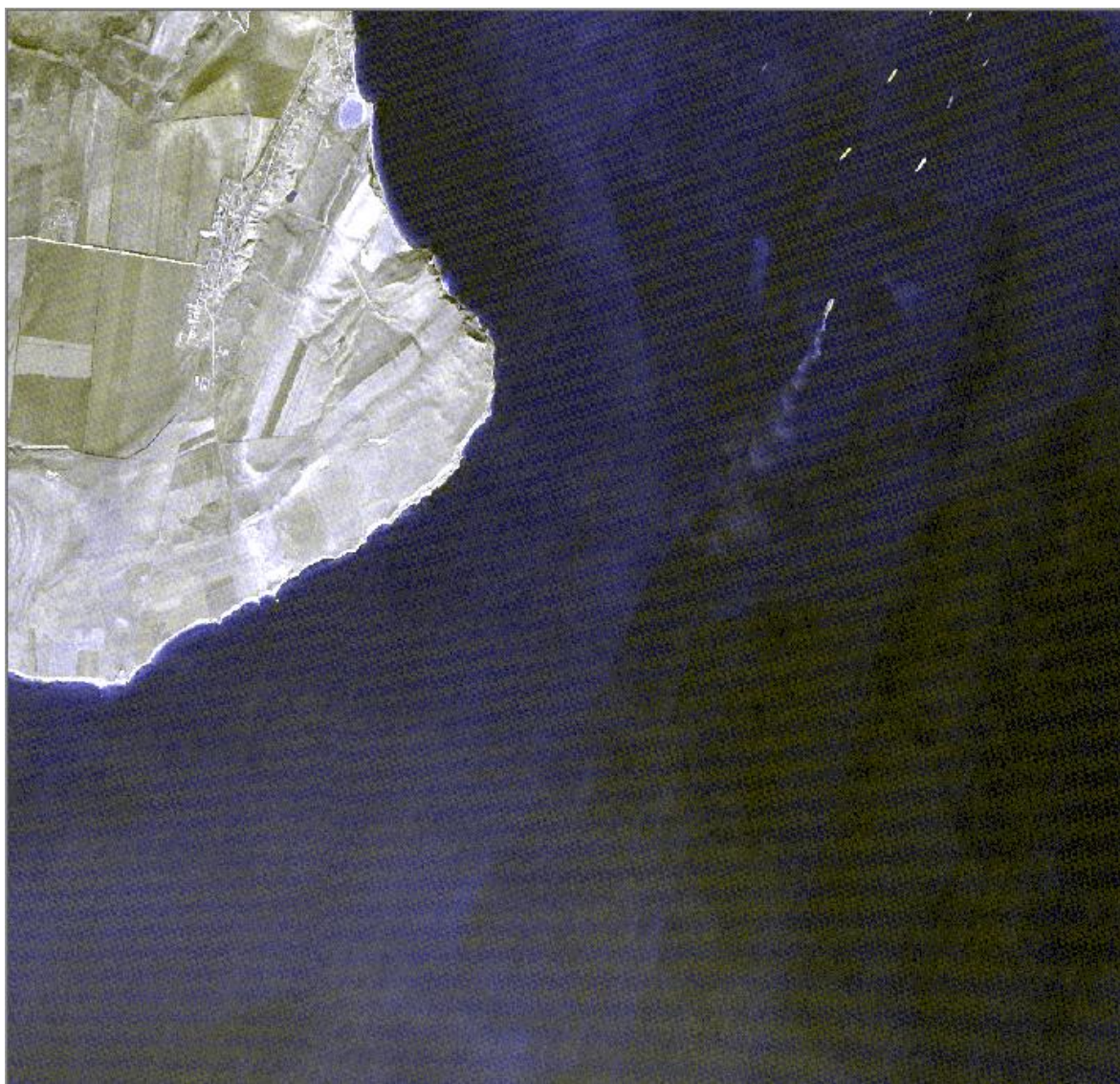


Рисунок 5.6 – знімок до обробки

Після застосування фільтру, фрагмент знімку прийняв наступний вигляд

(Рисунок 5.7). На знімку до обробки можна спостерігати шум, що досить сильно заважає візуально визначити межі нафтових розливів на морській поверхні.

Набір фільтрів в даній роботі був створений, насамперед, для усунення зайвих шумів.



Рисунок 5.7 – знімок після обробки

На рисунку 5.7 видно, що зображення дещо розмилося і стало краще видно межі забруднення. При більш ретельній обробці (більшій кількості застосованих фільтрів), межі нафтового забруднення буде видно досить чітко.

Висновки до розділу

Отже, було розроблено систему моніторингу нафтових розливів на морській поверхні. Виходячи з цього розділу, можна зробити висновок, що дана має певні переваги перед існуючими аналогами:

- простота та швидкість виявлення плям
- широкі можливості налаштування інтерфейсу;
- інтуїтивно зрозумілий інтерфейс;
- можливість без зайвих зусиль додати, редагувати або видалити космічні

знімки.

Завдяки цьому розділу, будь-який користувач без зайвих зусиль зможе провести моніторинг нафтових розливів на морській поверхні.

ВИСНОВКИ

Оскільки моніторинг нафтових розливів на морській поверхні є досить важливою задачею для фахівців, що займаються спостереженням та дослідженням акваторій та екологів, було розроблено систему у вигляді додатку до ArcMap із зручним інтерфейсом, яка надає можливість визначати межі нафтових розливів на морській поверхні та нафтових плям, шляхом обробки готових радіолокаційних та космічних знімків. Також, важливою причиною для розробки подібної системи являється необхідність обробляти великі масиви даних.

До переваг даної системи можна віднести:

- простота та швидкість виявлення плям
- широкі можливості налаштування інтерфейсу;
- інтуїтивно зрозумілий інтерфейс;
- можливість без зайвих зусиль додати, редагувати або видалити космічні

знімки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Довгий С.О., Лялько В.І., Бабійчук С.М. Основи дистанційного зондування Землі: історія та практичне застосування, - Київ, 2019 – 316 с.
2. ДеМерс Майкл Н. Географические информационные системы. Основы.: Пер. с англ. – М.: Дата+, 1999. – 490 с.
3. Василега В.Д. Ландшафтна екологія: Навч. — Суми: Вид-во СумДУ, 2010. — 303 с.
4. Lee, J.S. 1980. Digital image enhancement and noise filtering by use of local statistics. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-2, no. 2, pp. 165–168.
5. Шовенгердт Р.А. Дистанционное зондирование. Методы и модели и методы обработки изображений.
6. Эндрю Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5 /Эндрю Троелсен. – Москва : Вильямс, 2013, — 1311 с.
7. Джеффри Рихтер. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Джеффри Рихтер. – Санкт-Петербург : Питер, 2017, — 896 с.
8. Герберт Шилдт. C# 4.0. Полное руководство / Герберт Шилдт. – Москва : Вильямс, 2011. — 1056 с.
9. Jon Skeet. C# in Depth. Second Edition / Jon Skeet, - NY : Manning Publications, 2008. - 554 с.
10. Ben Watson. Writing High-Performance .Net Code / Ben Watson, Leticia Watson - NY : Manning Publications, 2009. – 683 с.
11. Бахвалов Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. — Москва: Бином, 2001. — 363 с.
12. Боровков А. И. Компьютерный инжиниринг / А. И. Боровков. — СПб: Изд-во Политехн. ун-та, 2012. — 93 с.

13. Майк Гандерлой. Освоение Microsoft SQL Server 2005 / Майк Гандерлой, Джозеф Джорден, Дейвид Чанц. — М.: «Диалектика», 2007. — 1104 с.
14. Bill Hamilton. ADO.NET 3.5 Cookbook, 2nd Edition. Building Data-Centric .NET Applications / Bill Hamilton, - CA : O'Reilly Media, 2008, - 982 с.
15. Julia Lerman. Programming Entity Framework, 2nd Edition / Julia Lerman, - CA: O'Reilly Media, 2010, - 920с.

ДОДАТОК 1

Моніторинг нафтових розливів на морській поверхні методами
дистанційного зондування Землі

Специфікація

УКР.НТУУ «КПІ ім. І.Сікорського». ТМ5213_19Б

Аркушів 2

| Позначення | Найменування | Примітки |
|--|--|---|
| Документація | | |
| УКР.НТУУ «КПІ ім. І.Сікорського». ТМ5213_19Б 81-1 | Записка_Панченко_Т М52.docx | Пояснювальна записка |
| Компоненти | | |
| УКР.НТУУ «КПІ ім. І.Сікорського». ТМ5213_19Б 12-01 | ImageFilterProject\I mageFilterProject/ImageF ilter.cs | Клієнтський застосунок з усіма компонентами |
| УКР.НТУУ «КПІ ім. І.Сікорського». ТМ5213_19Б 12-02 | ImageFilterProject\I mageFilterProject/App.con fig | Файл конфігурації |

ДОДАТОК 2

Моніторинг нафтових розливів на морській поверхні методами
дистанційного зондування Землі

Текст програмного модуля

УКР.НТУУ «КПІ ім. І.Сікорського». ТМ5213_19Б

Аркушів 7

```

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using ESRI.ArcGIS.ArcMapUI;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.DataSourcesRaster;
using ESRI.ArcGIS.Geodatabase;
using ESRI.ArcGIS.Geometry;

namespace ImageFilterProj
{
    public class ImageFilter : ESRI.ArcGIS.Desktop.AddIns.Button
    {
        /*
         * Провайдер фільтрів, що використовується для збереження фільтрів
         */
        private Dictionary<string, double[,]> provider = new Dictionary<string, double[,]>
        {
            {"Gausse5", new double[5, 5]
            {
                { 0.00002292,0.00019117,0.00038771,0.00019117,0.00002292 },
                { 0.00002292,0.00078633,0.00655965,0.01330373,0.00655965 },
                { 0.00019117,0.00655965,0.05472157,0.11098164,0.05472157 },
                { 0.00019117,0.00038771,0.01330373,0.11098164,0.22508352 },
                { 0.01330373,0.00038771,0.00019117,0.00655965,0.05472157 }
            }
            },
            {"Gausse3", new double[3, 3]
            {
                { 0.00078633,0.00655965,0.01330373 },
                { 0.00655965,0.05472157,0.11098164 },
                { 0.00038771,0.01330373,0.11098164 }
            }
            },
        };

        public ImageFilter()
        {
        }

        /*
         * Основний метод, що викликається при натиску на кнопку
         */
        protected override void OnClick()
        {
            var k = provider["Gausse5"];

```

```

    ExecuteFilter(k);
    ArcMap.Application.CurrentTool = null;
}
/*
 * Метод, що відкриває знімок
 * і застосовує фільтр
 */
private static void ExecuteFilter(double[,] filter)
{
    //Взяти нульовий шар знімку
    IRasterLayer rLayer = GetLayer(0);
    IRaster2 rast2 = rLayer.Raster as IRaster2;
    //Кількість каналів
    int bandCount = rLayer.BandCount;
    //Кількість стовпчиків
    int columnCount = rLayer.ColumnCount;
    //Кількість рядків
    int rowCount = rLayer.RowCount;
    IWorkspaceFactory workspaceFact = new RasterWorkspaceFactory();
    IRasterWorkspace2 rasterWs = workspaceFact.OpenFromFile(@"F:\Gisi\Filter",
0) as IRasterWorkspace2;
    //Інформація про канали знімку
    ISpatialReference sr = new UnknownCoordinateSystem() as ISpatialReference;
    double mapX;
    double mapY;
    rast2.PixelToMap(0, 0, out mapX, out mapY);
    IPoint origin = new Point();
    origin.PutCoords(mapX, mapY);

    int width = columnCount;
    int height = rowCount;
    double xCell = 15;
    double yCell = 15;
    int NumBand = bandCount;
    /* Створення нового набору даних, що буде результирующим,
     * розмір і канали якого відповідають розміру і каналам,
     * вхідного знімку
     */
    string fileName = "g" + DateTime.Now.Millisecond;
    IRasterDataset rasterDataset = rasterWs.CreateRasterDataset(fileName, "GRID",
        origin, width, height, xCell, yCell, NumBand, rstPixelType.PT_FLOAT, sr,
        true);
    //Створення нового растру

```

```

IRaster raster = rasterDataset.CreateDefaultRaster();
IRasterBandCollection rasterBands = raster as IRasterBandCollection;
IRasterBand rasterBand;
IRasterProps rasterProps;
rasterBand = rasterBands.Item(0);
rasterProps = rasterBand as IRasterProps;
//Створення блоку пікселів
IPnt blocksize = new Pnt();
blocksize.SetCoords(width, height);
IPixelBlock3 pixelblock = raster.CreatePixelBlock(blocksize) as IPixelBlock3;
//Цикл, що попіксельно застосовує фільтр і
//заповнює новий знімок значеннями
for (int i = 0; i < bandCount; i++)
{
    float[,] pixels = pixelblock.PixelData[i];
    for (int col = 0; col <= columnCount; col++)
    {
        for (int row = 0; row <= rowCount; row++)
        {
            //Накладення фільтру
            pixels[col, row] = (float)GetFilteredPixel(rast2, filter, i, col, row,
rowCount, columnCount);
        }
    }
    pixelblock.PixelData[i] = pixels;
    IPnt upperLeft = new Pnt();
    upperLeft.SetCoords(0, 0);
    //Оновлення результуючого знімку
    IRasterEdit rasterEdit = raster as IRasterEdit;
    rasterEdit.Write(upperLeft, pixelblock as IPixelBlock);
}

}
/*
 * Метод доступу до растрових даних, що повертає растр
 */
private static IRasterLayer GetLayer(int layer)
{
    IMxDocument pMxDoc = ArcMap.Application.Document as IMxDocument;
    IMap pMap = pMxDoc.FocusMap;
    IRasterLayer rLayer = pMap.Layer[layer] as IRasterLayer;
    return rLayer;
}
/*

```

```

* Метод накладення фільтру
*/
private static double GetFilteredPixel(IRaster2 rast, double[,] filter, int band, int
column, int row, int rastLenght, int rastWidth)
{
    int filterLenght = filter.GetLength(0);
    int filterWidth = filter.GetLength(1);

    int startRow = row - filterLenght / 2;
    int startColumn = column - filterWidth / 2;
    double sum = 0;

    for (int i = 0; i < filterLenght; i++)
    {
        for (int j = 0; j < filterWidth; j++)
        {
            int pI = Math.Abs(startRow + i);
            pI = pI > rastLenght - 1 ? rastLenght - 1 - (pI + 1 - rastLenght) : pI;
            int pJ = Math.Abs(startColumn + j);
            pJ = pJ > rastWidth - 1 ? rastWidth - 1 - (pJ + 1 - rastWidth) : pJ;
            sum += filter[i, j] * Convert.ToDouble(rast.GetPixelValue(band, pJ, pI));
        }
    }

    return sum;
}

protected override void OnUpdate()
{
    Enabled = ArcMap.Application != null;
}
}

using System;
using System.Data.Entity;

namespace ImageFilterProj
{

    public class Filter
    {
        public System.Guid Id { get; set; }
        public string Name { get; set; }
    }
}

```

```

    public string Body { get; set; }
}

public class FilterContext : DbContext
{
    public FilterContext()
        : base("name=mssql_db")
    {
    }
    public virtual DbSet<Filter> Filters { get; set; }
}
}

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit
http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,
EntityFramework, Version=6.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" requirePermission="false" />
  </configSections>
  <connectionStrings>
    <add name="mssql_db" connectionString="Data
Source=(localdb)\MSSQLLocalDB;Initial Catalog=ImageFilters;Integrated
Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False"
providerName="System.Data.SqlClient" />
  </connectionStrings>
  <entityFramework>
    <defaultConnectionFactory
type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">
      <parameters>
        <parameter value="mssqllocaldb" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
      <provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    </providers>
  </entityFramework>
</configuration>

```

ДОДАТОК 3

Моніторинг нафтових розливів на морській поверхні методами
дистанційного зондування Землі

Опис програмного модуля

Аркушів 10

АНОТАЦІЯ

Програмний продукт являє собою додаток-кнопку до ArcMap, створений мовою програмування C# на платформі Microsoft .Net з використанням фреймворків ADO.Net, Entity Framework. Розробка інтерфейсу додатку відбувалася за допомогою API WinForms та ArcObject SDK. В якості середовища розробки – Microsoft Visual Studio 2015 Enterprise. Додаток може використовуватися науковцями та екологами, при проведенні спостережень морської поверхні та оцінки забруднення.

ЗМІСТ

| | |
|--|----|
| 1. Загальні відомості..... | 4 |
| 2. Функціональне призначення | 5 |
| 3. Опис логічної структури..... | 6 |
| 4. Використовувані технічні засоби | 7 |
| 5. Виклик і завантаження..... | 8 |
| 6. Вхідні дані | 9 |
| 7. Вихідні дані..... | 10 |

1. ЗАГАЛЬНІ ВІДОМОСТІ

Серверна частина програмного продукту " Моніторинг нафтових розливів на морській поверхні методами дистанційного зондування Землі" є - десктопним додатком до ArcMap, написаним мовою програмування C#.

Клієнтський інтерфейс написаний мовою програмування C# з використанням API WinForms та ArcObject SDK.

Для функціонування системи необхідний комп'ютер з операційною системою Windows 10 з встановленим ArcGis 10.4.1.

2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Система " Моніторинг нафтових розливів на морській поверхні методами дистанційного зондування Землі" призначена для науковців та екологів, які проводять різноманітні спостереження морських поверхонь.

Не менш важливою складовою є те, що даний застосунок був розроблений з метою зробити його користування для користувача якомога простішим та зручнішим.

3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Система " Моніторинг нафтових розливів на морській поверхні методами дистанційного зондування Землі" працює методом накладання ряду специфічних фільтрів. Тобто, програма попіксельно накладає на вхідний знімок фільтри і формує вихідне зображення.

4. ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для функціонування системи " Моніторинг нафтових розливів на морській поверхні методами дистанційного зондування Землі" необхідний будь-який комп'ютер з ОС Windows, на якому встановлений ArcGis 10.4.1.

5. ВИКЛИК І ЗАВАНТАЖЕННЯ

Для встановлення програми на новий комп'ютер необхідно (приклад для ОС Windows 10):

- скопіювати папку з програмою на комп'ютер;
- встановити ArcGis 10.4.1;
- змінити в ImageFilterProject\ImageFilterProject\ImageFilter.cs налаштування доступу до робочого простору;

6. ВХІДНІ ДАНІ

— космічний або радіолокаційний знімок;

7. ВИХІДНІ ДАНІ

Вихідною інформацією є:

— нове растрове зображення, на якому можна візуально побачити наявність та межі нафтових розливів;